

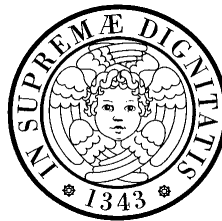
*Alla mia famiglia
e ai miei amici.
In memoria di mio padre
e dei miei nonni.*

UNIVERSITÀ DEGLI STUDI DI PISA

Facoltà di Ingegneria
Corso di Laurea in Ingegneria Aerospaziale

Tesi di Laurea

**Sviluppo e validazione sperimentale di un
filtro di Kalman applicato alla navigazione
inerziale di velivoli**



Relatori:
Prof. Giovanni Mengali

Laureando:
Daniele SENI

Prof. Roberto Galatolo

Correlatore:
Ing. Francesco Schettini

04 MV 02 521

ANNO ACCADEMICO 2002-2003

Sommario

La presente tesi si inquadra nel progetto di ricerca denominato “SCAUT” (Sistema di Controllo AUtomatico del Territorio) che il DIA (Dipartimento di Ingegneria Aerospaziale) e il DIASP (Dipartimento di Ingegneria Aeronautica e Spaziale del Politecnico di Torino) stanno sviluppando allo scopo di realizzare un velivolo non abitato (UAV-*Uninhabited Aerial Vehicle*) per scopi civili.

In particolare questo lavoro si occupa della programmazione in ambiente *Simulink* del sistema di navigazione inerziale di bordo (INS-*Inertial Navigation System*) e del *filtro di Kalman* necessario ad eliminare gli errori causati dall’integrazione del rumore associato alle accelerazioni e alle velocità angolari provenienti da una strumentazione di tipo AHRS (Attitude & Heading Reference System) basata su sensori MEMS (Micro Electro Mechanical Systems). Il modello è stato inoltre strutturato in maniera tale da prevedere l’utilizzo di ricevitori GPS (*Global Position System*) per il riallineamento dell’assetto e la correzione della traiettoria di volo.

La strumentazione sperimentale prevede infine una struttura *hardware* di tipo *host/client* in rete *ethernet* per l’implementazione in tempo reale mediante il *toolbox di Matlab®* denominato “*xPC Target*”.

Ringraziamenti

Desidero ringraziare sentitamente tutte le persone che hanno collaborato alla realizzazione di questo lavoro.

In particolare un sincero grazie v  ai Proff. G. Mengali e R. Galatolo per i preziosi consigli, la pazienza e la costante presenza dimostratemi nello svolgimento di questa prova.

Fondamentale e insostituibile   stato il supporto dell'Ing. F. Schettini, col quale   nato anche un buon rapporto di amicizia e dell'Ing. A. Quarta per i preziosi consigli in fase di stesura della relazione.

Questa tesi deve la sua realizzazione anche grazie all'aiuto dell'Ing. M. Pisani, del Sig. M. Bernardini e del personale del Laboratorio di Elettronica, che con competenza e disponibilit  hanno realizzato gran parte della strumentazione sperimentale.

Non potevo infine dimenticare gli amici ed i colleghi che mi hanno sostenuto moralmente nel corso degli anni. Un particolare abbraccio   dedicato a Cinzia e a mia madre che nonostante i miei sbalzi d'umore e i periodi difficili hanno saputo mantenere inalterato il loro affetto e la loro attenzione.

Indice

Sommario	I
Ringraziamenti	II
Elenco dei simboli	VI
I Navigazione inerziale	1
1 Introduzione	2
1.1 Scopo della tesi	2
1.2 Fasi della tesi	3
1.3 Architettura del modello	4
1.4 Architettura hardware	5
2 Modello matematico non lineare	9
2.1 Sistemi di riferimento	9
2.2 Equazioni di moto del velivolo	10
2.3 Equazioni di assetto del velivolo	12
2.3.1 Generalità	12
2.3.2 Modello matematico dei quaternioni	13
2.4 Struttura Simulink del modello matematico	17
2.4.1 Struttura Blocco Accelerometri	17
2.4.2 Struttura Blocco Giroscopi	23
2.5 Esempio di simulazione	25

3	Modello matematico linearizzato	27
3.1	Linearizzazione delle equazioni di moto e di assetto	27
3.2	Struttura Simulink del modello matematico	30
3.3	Esempio di simulazione	39
4	Filtro di Kalman	42
4.1	Generalità	42
4.2	Richiami sui segnali aleatori	43
4.2.1	Funzioni caratteristiche dei processi aleatori	43
4.2.2	Rumore “bianco”	44
4.3	Modello matematico	45
4.4	Struttura Simulink del filtro di Kalman	48
4.4.1	Blocco <i>GPS</i>	50
4.4.2	Blocco <i>n-Hz</i>	50
4.4.3	Blocco <i>Filtro Kalman</i>	51
4.5	Esempi di simulazione	54
II	Risultati sperimentali	61
5	Attività sperimentale	62
5.1	Introduzione	62
5.2	Preparazione dei test	62
5.2.1	Test con ingressi simulati	62
5.2.2	Test “ <i>Real Time</i> ” con ingressi reali	63
5.3	Descrizione esperimento	64
5.4	Elaborazione dei dati	65
6	Conclusioni e sviluppi futuri	70
6.1	Conclusioni	70
6.2	Sviluppi futuri	70
A	Linearizzazione	72
B	Approfondimento sulla S-Function	75
C	Settaggio di “<i>xPC Target</i>”	77

Bibliografia	80
Elenco delle figure	81
Elenco degli acronimi	84

Elenco dei Simboli

α, β, γ	Coseni direttori asse istantaneo di rotazione
η	Angolo di rotazione attorno all'asse istantaneo di rotazione
λ	Angolo di latitudine
μ	Media
ω_s	Velocità angolare terrestre
ϕ, θ, ψ	Angoli di Eulero (assetto velivolo)
ρ	Funzione di correlazione
σ^2	Varianza
φ	Angolo di longitudine
$\hat{\mathbf{x}}$	Vettore degli stati stimati
\mathbf{A}	Matrice jacobiana degli stati
\mathbf{B}	Matrice jacobiana degli ingressi
\mathbf{e}	Vettore errore sulle stime
\mathbf{H}	Matrice jacobiana delle misurazioni
\mathbf{K}	Matrice dei guadagni di Kalman
\mathbf{P}	Matrice di covarianza dell'errore
\mathbf{u}	Vettore degli ingressi

\mathbf{V}	Matrice di covarianza rumore degli stati
\mathbf{v}	Vettore del rumore bianco associato agli stati
\mathbf{W}	Matrice di covarianza rumore delle misurazioni
\mathbf{w}	Vettore del rumore bianco associato alle misurazioni
\mathbf{x}	Vettore degli stati
\mathbf{X}_b	Vettore generico espresso in assi corpo
\mathbf{X}_v	Vettore generico espresso in assi verticali locali
\mathbf{z}	Vettore delle misurazioni
ξ	Parametro di correzione errori nella dinamica dei quaternioni
$A_{N,E,V}$	Componenti accelerazione baricentro velivolo in assi verticali locali
$A_{x,y,z}$	Accelerazioni del baricentro del velivolo in assi corpo
$E[...]$	Valore atteso o media in senso probabilistico
$E_{0,1,2,3}$	Parametri di Eulero
$f_X(x,t)$	Funzione densità di probabilità
g	Accelerazione di gravità
H	Quota di volo
p,q,r	Velocità angolari velivolo espresse in assi corpo
R	Somma del raggio terrestre e della quota di volo
T_{sim}	Tempo di simulazione in ambiente <i>Simulink</i>
$V_{N,E,V}$	Componenti velocità baricentro velivolo in assi verticali locali

Parte I

Navigazione inerziale

1

Introduzione

1.1 Scopo della tesi

Questo lavoro ha lo scopo di sviluppare un'applicazione *software* per l'integrazione numerica delle equazioni di moto di un generico velivolo, avendo come dati di ingresso tre accelerazioni e tre velocità angolari provenienti da una piattaforma inerziale. Tali segnali sono ovviamente affetti da “rumore” (che assumeremo per ipotesi “bianco”); la loro integrazione determina una deriva e quindi un errore sulle quantità da esse ricavate.

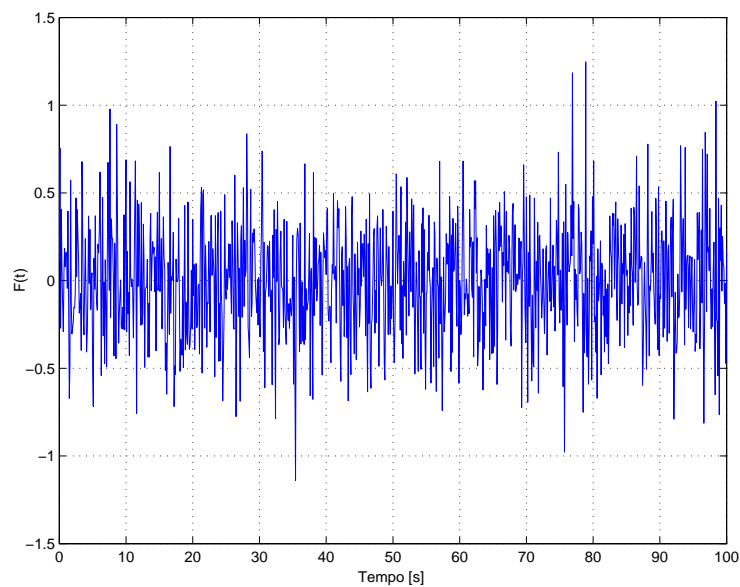


Figura 1.1: Esempio di *rumore bianco*

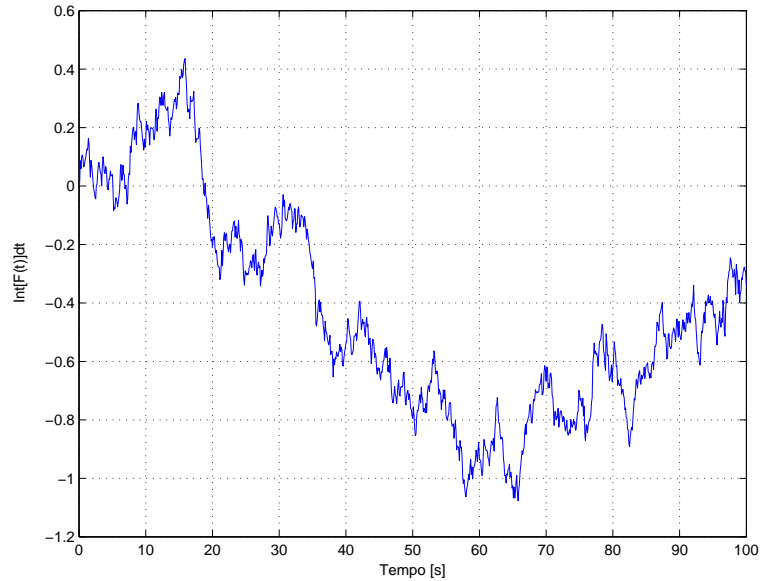


Figura 1.2: Esempio di integrazione di *rumore bianco*

A titolo di esempio si riportano i risultati di una semplice simulazione (vedi Figure 1.1 e 1.2): il segnale di Figura 1.1 è dato dalla sovrapposizione di una costante nulla (ingresso nominale) con del rumore bianco, mentre la Figura 1.2 rappresenta il risultato della sua integrazione. Si nota chiaramente da quest'ultima un deterioramento progressivo.

In una situazione del genere appare logica la necessità di utilizzare uno strumento matematico che ci permetta di eliminare tali derive; nello specifico si è scelto di utilizzare, fra tutti i filtri a disposizione in letteratura, quello di Kalman [1] che ci consente di confrontare una grandezza “predetta” (calcolata) con una “misurata” direttamente mediante opportuni strumenti diversi dalla piattaforma.

Al fine di validare la bontà del modello si è proceduto infine ad una serie di sperimentazioni con simulazione in tempo reale e non.

1.2 Fasi della tesi

Nella prima fase del lavoro svolto è stata realizzata una ricerca bibliografica che ha permesso di capire il funzionamento delle piattaforme inerziali e di mettere in luce le problematiche connesse al sistema di navigazione di un velivolo e le metodologie per ridurre gli errori causati dal rumore associato

alle misurazioni.

Il materiale raccolto ha consentito di procedere con l'implementazione in ambiente *Matlab* (mediante un suo *toolbox* denominato *Simulink*) dei modelli “non lineari” relativi alla dinamica del velivolo.

Una volta validati tali modelli attraverso alcune simulazioni, si è passati a “linearizzare” la dinamica del moto del velivolo al fine di poter utilizzare il “*Filtro di Kalman*” (i filtri di Kalman si basano infatti sulla linearità della dinamica da ricostruire). Anche in questo caso prima di proseguire col lavoro si è proceduto col validare, tramite opportune simulazioni, il modello “linearizzato” paragonando i risultati ottenuti con quelli derivanti dal modello “non lineare” precedente.

Infine si è realizzato il programma relativo al filtro, testato mediante introduzione di “rumore bianco” agli ingressi del sistema.

Parallelamente alla realizzazione del *software* è stata assemblata una strumentazione *hardware* simile a quella che sarà installata a bordo del velivolo. Essa, nella fase di sperimentazione finale, è stata montata a bordo di un'auto in movimento al fine di acquisire i dati dalla piattaforma inerziale e dal GPS, svolgendo inoltre il ruolo di *target* nelle prove in *real time*. Tali dati sono stati poi elaborati in *post-process* al fine di ricostruire le grandezze del moto (velocità, posizione, ecc).

1.3 Architettura del modello

Il filtro realizzato è composto essenzialmente da tre blocchi principali (vedi Figura 1.3):

- il blocco INS, in cui vengono integrate le equazioni della meccanica del volo (“predizioni”);
- Il blocco GPS, che provvede alle misurazioni da confrontare con le quantità analoghe predette dall'INS;
- Il filtro Kalman, che permette di ricostruire i segnali deteriorati dovuti all'integrazione degli ingressi affetti da rumore.

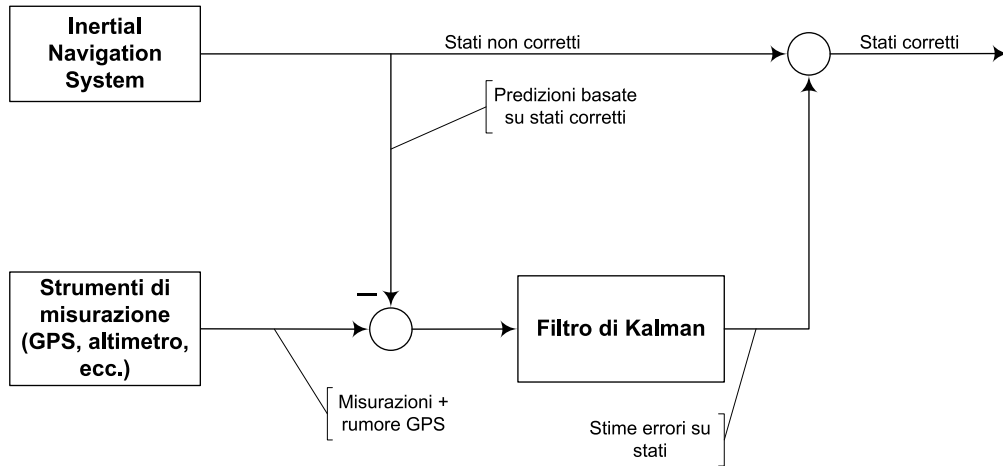


Figura 1.3: Schema dell'architettura del filtro

1.4 Architettura hardware

La sperimentazione finale è stata possibile grazie ad un hardware costituito dai seguenti componenti (vedi Figura 1.4):

- Piattaforma inerziale *Crossbow* tipo AHRS400cc-200
- Scheda di acquisizione dati e convertitore A/D *Diamond* M-16AT con architettura PC-104;
- Single Board Computer LS-560 prodotto dalla *Commell Industrial Computer*, svolgente il ruolo di “target” nella simulazione in *Real Time*, con interfaccia PC-104 (104mm x 104mm x 20mm) e scheda di rete integrata;
- Personal Computer su cui gira il modello *simulink*, svolgente il ruolo di “server” nella simulazione in *Real Time*.

Vale la pena di sottolineare che il “Computer LS-520” di cui sopra non sarà installato realmente a bordo del velivolo; infatti esso ha una potenza di calcolo che probabilmente risulterà eccessiva per l’INS. Al fine di ridurre ingombri, costi e consumi energetici dopo un’attenta fase di calibrazione si utilizzerà perciò anche per il computer di bordo una architettura PC-104. In poche parole si è acquistata la scheda di acquisizione definitiva rimandando la scelta del computer miniaturizzato alla fine dei test di valutazione.

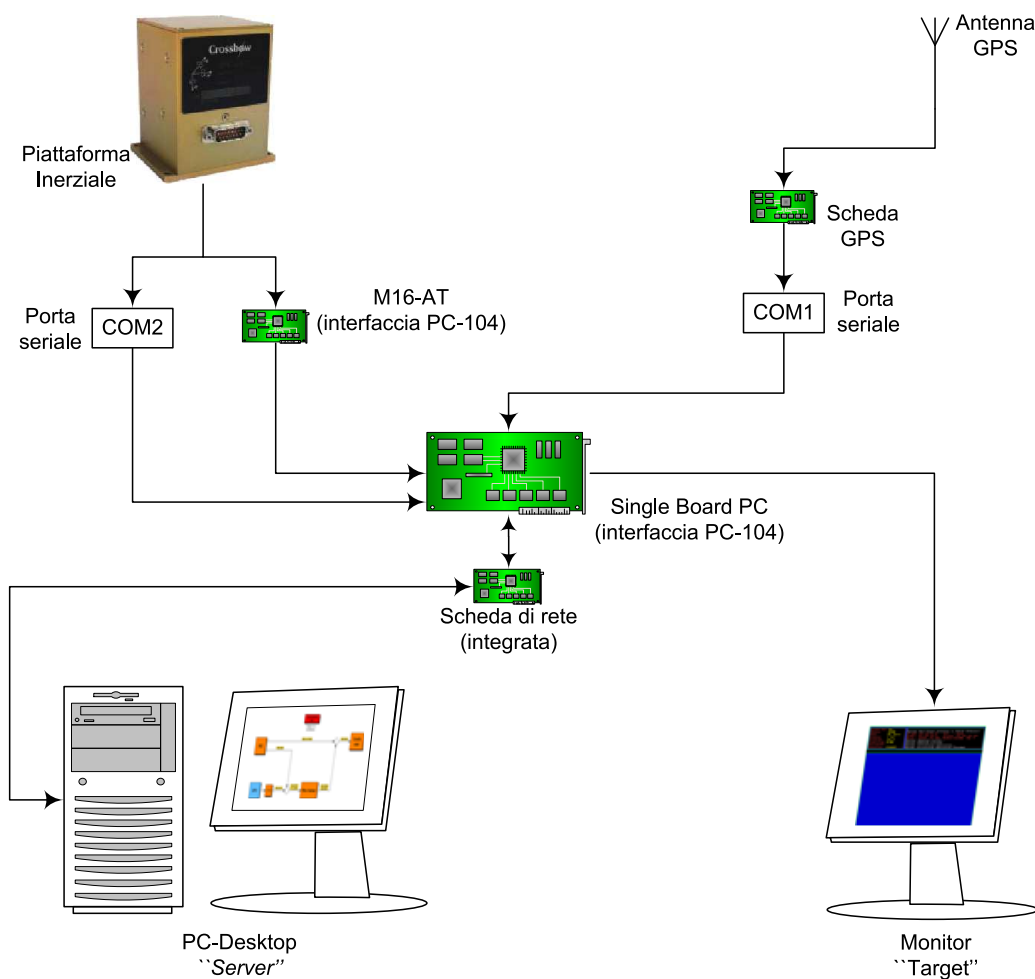


Figura 1.4: Schema dell'architettura hardware

La piattaforma inerziale utilizzata è di tipo integrato in quanto contiene al suo interno tre accelerometri (lungo tre assi ortogonali), tre giroscopi e tre magnetometri e sfrutta le metodologie costruttive “*strapdown*”. Con questo nome si identificano le piattaforme prive di giroscopi rotanti che, utilizzando le moderne tecnologie denominate MEMS (*Micro Electro Mechanical Systems*), percepiscono le velocità angolari del mezzo su cui sono montate sfruttando l'effetto *Coriolis*.

L'acquisizione di accelerazioni, velocità angolari ed eventualmente di campi magnetici può essere fatta in vari modi, programmando opportunamente il modo di lavoro tra quelli disponibili che sono:

“**Angle mode**” in cui le accelerazioni sono in numeri di g , le velocità angolari in gradi al secondo e i campi magnetici in *gauss*. Questo modo calcola internamente anche gli angoli di assetto.

“**Scaled sensor mode**” simile al precedente (a parte gli angoli che non sono disponibili) con in più delle correzioni interne dovute a disallineamenti e variazioni di temperatura

“**Voltage mode**” in cui si hanno le stesse uscite del precedente (non corrette) però espresse in *Volts*. La conversione in unità ingegneristiche è poi possibile tramite leggi empiriche reperibili dal manuale allegato alla strumentazione.

Tali uscite sono disponibili sia in formato digitale che analogico (vedi Figura 1.5); nel primo caso l'acquisizione dei dati avviene tramite porta seriale mentre il secondo necessita della scheda (con interfaccia PC-104) “Diamond M16-AT” (vedi Figura 1.4).

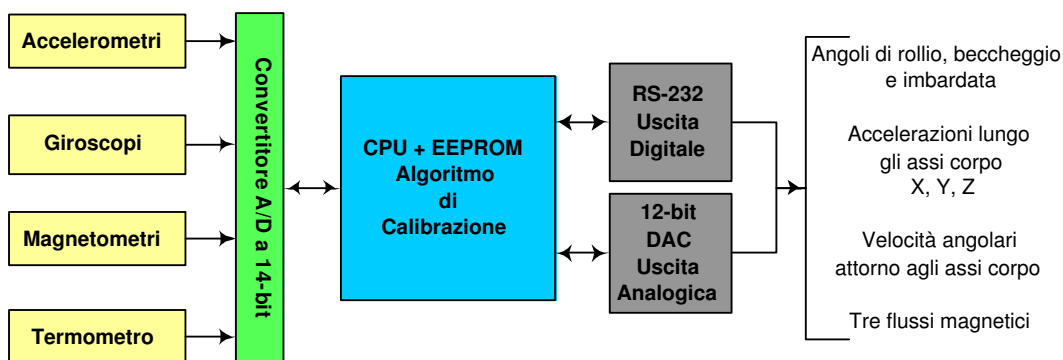


Figura 1.5: Schema elettronico AHRS

Le caratteristiche principali di efficacia e accuratezza della piattaforma sono di seguito riassunte:

Velocità di aggiornamento dati	$\approx 60 \text{ Hz}$
Tempo di warm-up	$\approx 60 \text{ s}$
Temperatura di esercizio	$-40 \div 71 \text{ }^{\circ}\text{C}$

Accelerometri

Massima accelerazione rilevabile	$\pm 10 \text{ g}$
Risoluzione	$< 0.05 \text{ }^{\circ}/\text{s}$
Banda passante	$> 25 \text{ Hz}$
Range di tensione uscite analogiche	$0 \div 5 \text{ V}$

Giroscopi

Massima velocità angolare rilevabile	$\pm 200 \text{ }^{\circ}/\text{s}$
Risoluzione	$< 1.25 \text{ mg}$
Banda passante	$> 10 \text{ Hz}$
Range di tensione uscite analogiche	$\pm 4.096 \text{ V}$

Un dettaglio degno di nota è il seguente: con la piattaforma appoggiata su un tavolo in posizione orizzontale (asse z rivolto verso il basso), gli accelerometri lungo x e y rilevano un segnale medio pari a zero mentre quello lungo z rileva $1g$. In altre parole si avrebbe accelerazione pari a zero anche lungo z se la piattaforma precipitasse nel vuoto mantenendo tale asse costantemente verticale rivolto verso il basso.

2

Modello matematico non lineare

2.1 Sistemi di riferimento

I sistemi di riferimento usati per la definizione delle equazioni di moto del velivolo sono quelli classici della meccanica del volo [2]:

Inerziale con origine geocentrica, asse X diretto verso la costellazione dell'Ariete, asse Z passante per il polo nord geografico e asse Y tale da rendere la terna levogira (si trascura ovviamente il moto di rivoluzione della Terra attorno al Sole che per tempi sufficientemente lunghi può essere ritenuto rettilineo ed uniforme).

Terrestre avente origine e asse Z coincidenti col precedente, X passante per il meridiano di *Greenwich* e Y tale da rendere la terna levogira.

Verticale-Locale con origine nel baricentro del velivolo, Z orientato verso il basso secondo la direzione del vettore gravità locale, X e Y giacenti su un piano tangente alla superficie terrestre (o su uno ad esso parallelo se la quota di volo è maggiore di zero) dove X punta verso il polo nord magnetico e Y verso est.

Assi corpo con origine nel baricentro, X coincidente con l'asse longitudinale del velivolo, Z appartenente al piano longitudinale e puntato verso il basso (in condizioni di volo orizzontale) e asse Y tale da rendere la terna levogira.

2.2 Equazioni di moto del velivolo

Il punto di partenza di questo paragrafo è la prima equazione cardinale della meccanica newtoniana, che come è noto vale se riferita ad un sistema inerziale (come definito nel §2.1). Per esprimere tale equazione rispetto al sistema geografico che a noi interessa, si deve tener conto della velocità angolare terrestre (ipotesi di Terra ruotante) e di quella conseguente al moto attorno alla Terra (ipotesi di Terra sferica).

L'equazione vettoriale (espressa in coordinate geografiche), che dopo semplici passaggi algebrici ne deriva, è la seguente [3]:

$$\mathbf{A} + \mathbf{G} = \ddot{\mathbf{R}} + \boldsymbol{\varpi} \wedge \mathbf{R} + 2\boldsymbol{\varpi} \wedge \dot{\mathbf{R}} + \boldsymbol{\varpi} \wedge (\boldsymbol{\varpi} \wedge \mathbf{R}) \quad (2.1)$$

dove

$$\mathbf{A} = \begin{bmatrix} A_N \\ A_E \\ A_V \end{bmatrix} \quad \text{Vettore delle accelerazioni;}$$

$$\mathbf{R} = \begin{bmatrix} 0 \\ 0 \\ R \end{bmatrix} \quad \text{Vettore posizione rispetto al centro terrestre;}$$

$\boldsymbol{\varpi}$ Vettore velocità angolare assoluta in assi geografici;

\mathbf{G} Vettore accelerazione gravitazionale

L'ultimo termine a destra della (2.1) può essere riscritto come segue:

$$\boldsymbol{\varpi} \wedge (\boldsymbol{\varpi} \wedge \mathbf{R}) = \boldsymbol{\omega}_s \wedge (\boldsymbol{\omega}_s \wedge \mathbf{R}) + \begin{bmatrix} 2\omega_s V_E \sin \lambda + \frac{V_E^2}{R} \tan \lambda \\ -\omega_s V_N \sin \lambda - \frac{V_N V_E}{R} \tan \lambda \\ 2\omega_s V_E \cos \lambda + \frac{V_N^2 + V_E^2}{R} \end{bmatrix} \quad (2.2)$$

dove il primo addendo rappresenta l'accelerazione centripeta dovuta alla rotazione terrestre, avente la seguente espressione:

$$\boldsymbol{\omega}_s \wedge (\boldsymbol{\omega}_s \wedge \mathbf{R}) = \begin{bmatrix} R\omega_s^2 \cos \lambda \sin \lambda \\ 0 \\ R\omega_s^2 \cos^2 \lambda \end{bmatrix}, \quad \boldsymbol{\omega}_s = \begin{bmatrix} \cos \lambda \\ 0 \\ -\sin \lambda \end{bmatrix} \omega_s \quad (2.3)$$

Il vettore gravità può essere ottenuto sottraendo al vettore accelerazione gravitazionale \mathbf{G} il contributo dato da (2.3):

$$\mathbf{g} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = \mathbf{G} - \boldsymbol{\omega}_s \wedge (\boldsymbol{\omega}_s \wedge \mathbf{R}) \quad (2.4)$$

mentre la derivata temporale della velocità angolare totale si specializza nella seguente:

$$\dot{\boldsymbol{\omega}} = \begin{bmatrix} -(\omega_s \sin \lambda) \left(\frac{V_N}{R} \right) + \frac{\dot{V}_E}{R} + \frac{V_E V_V}{R^2} \\ -\frac{\dot{V}_N}{R} - \frac{V_N V_V}{R^2} \\ \dot{\omega}_3 \end{bmatrix} \quad (2.5)$$

Sostituendo nella (2.1) i risultati trovati e riscrivendola in forma scalare otteniamo le classiche equazioni di navigazione della meccanica del volo espresse in assi verticali-locali:

$$\dot{V}_N = A_N - (2\omega_s \sin \lambda + \frac{V_E}{R} \tan \lambda) V_E + \frac{V_N V_V}{R} \quad (2.6)$$

$$\dot{V}_E = A_E + (2\omega_s \sin \lambda + \frac{V_E}{R} \tan \lambda) V_N + 2\omega_s V_V \cos \lambda + \frac{V_E V_V}{R} \quad (2.7)$$

$$\dot{V}_V = A_V + g - 2\omega_s V_E \sin \lambda - \frac{V_N^2 + V_E^2}{R} \quad (2.8)$$

$$\dot{\lambda} = \frac{V_N}{R} \quad (2.9)$$

$$\dot{\varphi} = \frac{V_E}{R \cos \lambda} \quad (2.10)$$

$$\dot{H} = -V_V \quad (2.11)$$

dove:

V_N	Velocità baricentro velivolo lungo l'asse Nord [m/s]
V_E	Velocità baricentro velivolo lungo l'asse Est [m/s]
V_V	Velocità baricentro velivolo lungo l'asse Verticale [m/s]
A_N	Accelerazione baricentro velivolo lungo l'asse Nord [m/s^2]
A_E	Accelerazione baricentro velivolo lungo l'asse Est [m/s^2]
A_V	Accelerazione baricentro velivolo lungo l'asse Verticale [m/s^2]
λ	Angolo di latitudine [$^\circ$]
φ	Angolo di longitudine [$^\circ$]
R	Somma del raggio terrestre ($R_E \approx 6.378 \cdot 10^6$ [m]) e della quota di volo (H [m])
g	Accelerazione di gravità terrestre [m/s^2]
ω_s	Velocità angolare terrestre ($\approx 7.272 \cdot 10^{-5}$ [$^\circ/s$])

2.3 Equazioni di assetto del velivolo

2.3.1 Generalità

Come già accennato in §1.4, la piattaforma inerziale utilizzata è di tipo *strap-down*. Operativamente questo significa che essa viene montata sul velivolo allineando gli assi sensibili degli accelerometri e dei giroscopi con la terna assi corpo. Durante le evoluzioni dell'aereo, non essendo presente un riallineamento meccanico della piattaforma con gli assi verticali locali (come invece avviene per le piattaforme *compensate*, montate su sospensioni cardaniche), è necessario effettuare tale procedura “computazionalmente” (da cui il nome di “*piattaforme analitiche*”). Si rende quindi necessario l'aggiornamento continuo dell'assetto a partire da una configurazione iniziale nota; tale operazione può essere fatta in vari modi, tutti con dei pregi e dei difetti.

I principali metodi utilizzati nella simulazione del volo sono due:

1. Angoli di Eulero
2. Parametri di Eulero

Il primo richiede l'integrazione di tre equazioni differenziali a fronte delle quattro necessarie al secondo; nonostante questo fatto, l'assenza di funzioni

trigonometriche in quest'ultimo rende più snelle le operazioni matematiche da effettuare. Quanto detto è valido se ci limitiamo alla trasformazione fra sistemi di riferimento rinunciando alla conoscenza diretta degli angoli di Eulero. Tale operazione infatti esige un certo numero in più di operazioni aritmetiche che sommate alle precedenti superano quelle dell'altro metodo [4]; questo fatto diventa in parte marginale se teniamo conto dell'elevata potenza di calcolo dei moderni calcolatori e della possibilità di valutare gli angoli solo durante le manovre (evitando di aggiornarli ad ogni passo).

Altra differenza importante sta nel fatto che il metodo degli “angoli di Eulero” presenta una discontinuità matematica per angoli di beccheggio pari 90° , fenomeno da non confondere col bloccaggio meccanico noto col nome di “*gimbal lock*” sperimentato dalle piattaforme compensate a tre sospensioni cardaniche. Tale difetto potrebbe essere deleterio per velivoli acrobatici o comunque con inviluppi di volo particolarmente spinti.

Facendo un bilancio generale fra pregi e difetti dei due metodi si è scelto di utilizzare il secondo in quanto più sicuro e stabile.

2.3.2 Modello matematico dei quaternioni

In letteratura si trovano varie interpretazioni matematiche dei quaternioni. La più comune e di facile comprensione è quella detta a “*quattro parametri*”, dei quali tre identificano un vettore unitario (e quindi una retta chiamata comunemente “*asse istantaneo di rotazione*”, vedi Figura 2.1) mentre il quarto rappresenta l'angolo di rotazione vero e proprio del corpo attorno a tale asse (Teorema di Eulero). La terna assi corpo ad esempio per sovrapporsi alla terna geografica dovrà ruotare di un certo angolo attorno a quell'asse di rotazione.

Tramite alcuni passaggi matematici, che per brevità espositiva trascuriamo, si arriva alla matrice di rotazione basata sui *parametri di Eulero*:

$$\begin{bmatrix} X_v \\ Y_v \\ Z_v \end{bmatrix} = \begin{bmatrix} E_0^2 + E_1^2 - E_2^2 - E_3^2 & 2(E_1E_2 - E_0E_3) & 2(E_1E_3 + E_0E_2) \\ 2(E_1E_2 + E_0E_3) & E_0^2 - E_1^2 + E_2^2 - E_3^2 & 2(E_2E_3 - E_0E_1) \\ 2(E_1E_3 - E_0E_2) & 2(E_2E_3 + E_0E_1) & E_0^2 - E_1^2 - E_2^2 + E_3^2 \end{bmatrix} \cdot \begin{bmatrix} X_b \\ Y_b \\ Z_b \end{bmatrix} \quad (2.12)$$

in cui il pedice “ v ” sta ad indicare la terna assi verticali-locali mentre il pedice “ b ” quella “*assi-corpo*”

L’aggiornamento dei parametri presenti all’interno della matrice si attua tramite integrazione delle equazioni (2.13), (2.14), (2.15) e (2.16):

$$\dot{E}_0 = -\frac{1}{2} (pE_1 + qE_2 + rE_3) + \xi E_0 \quad (2.13)$$

$$\dot{E}_1 = \frac{1}{2} (pE_0 + rE_2 - qE_3) + \xi E_1 \quad (2.14)$$

$$\dot{E}_2 = \frac{1}{2} (qE_0 - rE_1 + pE_3) + \xi E_2 \quad (2.15)$$

$$\dot{E}_3 = \frac{1}{2} (rE_0 + qE_1 - pE_2) + \xi E_3 \quad (2.16)$$

dove (vedi Figura 2.1)

$$E_0 = \cos(\eta/2) \quad E_1 = \alpha \sin(\eta/2) \quad (2.17)$$

$$E_2 = \beta \sin(\eta/2) \quad E_3 = \gamma \sin(\eta/2) \quad (2.18)$$

e dove

$$\xi = 1 - (E_0^2 + E_1^2 + E_2^2 + E_3^2) \quad (2.19)$$

è un termine correttivo che permette di eliminare gli errori numerici di integrazione delle precedenti equazioni differenziali.

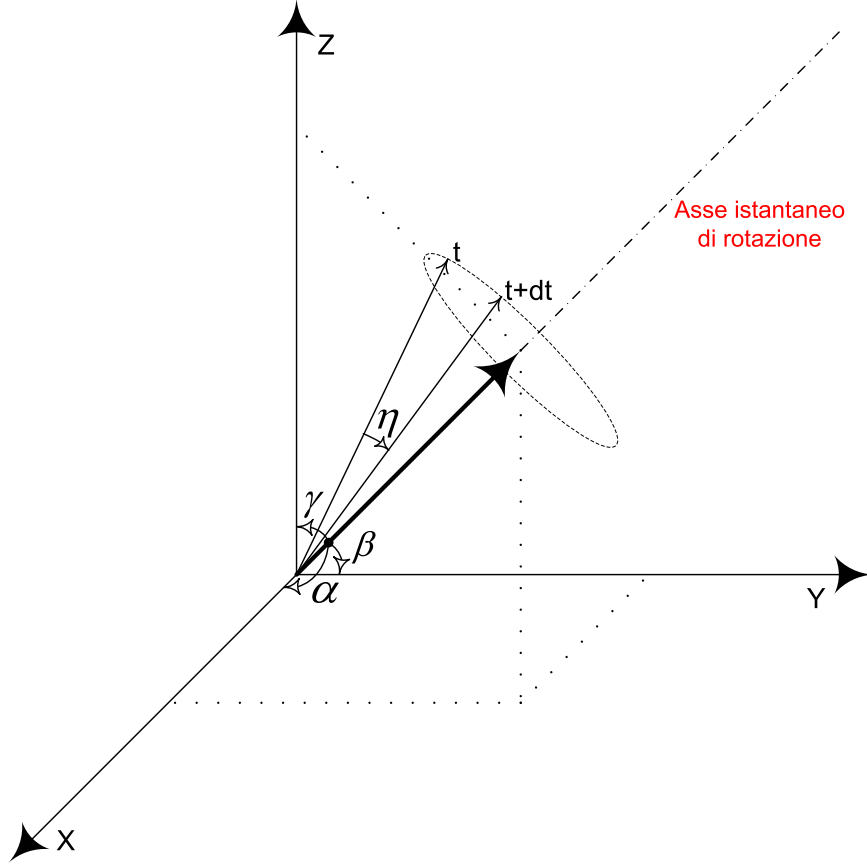


Figura 2.1: Definizione dei quaternioni

I valori iniziali per l'integrazione sono dati da:

$$E_0(0) = \cos \frac{\phi_0}{2} \cos \frac{\theta_0}{2} \cos \frac{\psi_0}{2} + \sin \frac{\phi_0}{2} \sin \frac{\theta_0}{2} \sin \frac{\psi_0}{2} \quad (2.20)$$

$$E_1(0) = \sin \frac{\phi_0}{2} \cos \frac{\theta_0}{2} \cos \frac{\psi_0}{2} - \cos \frac{\phi_0}{2} \sin \frac{\theta_0}{2} \sin \frac{\psi_0}{2} \quad (2.21)$$

$$E_2(0) = \cos \frac{\phi_0}{2} \sin \frac{\theta_0}{2} \cos \frac{\psi_0}{2} + \sin \frac{\phi_0}{2} \cos \frac{\theta_0}{2} \sin \frac{\psi_0}{2} \quad (2.22)$$

$$E_3(0) = \cos \frac{\phi_0}{2} \cos \frac{\theta_0}{2} \sin \frac{\psi_0}{2} - \sin \frac{\phi_0}{2} \sin \frac{\theta_0}{2} \cos \frac{\psi_0}{2} \quad (2.23)$$

dove ϕ_0, θ_0, ψ_0 sono gli angoli di assetto al tempo zero.

Gli angoli di Eulero vengono infine ricavati confrontando la matrice in (2.12) con l'analogica matrice di rotazione funzione degli angoli stessi qui riportata:

$$\begin{bmatrix} \cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \theta \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix} \quad (2.24)$$

ottenendo, dopo alcune semplificazioni:

$$\phi = \arctan \frac{2(E_0 E_1 + E_2 E_3)}{E_0^2 - E_1^2 - E_2^2 + E_3^2} \quad (2.25)$$

$$\theta = \arcsin [2(E_0 E_2 - E_1 E_3)] \quad (2.26)$$

$$\psi = \arctan \frac{2(E_0 E_3 - E_1 E_2)}{E_0^2 + E_1^2 - E_2^2 - E_3^2} \quad (2.27)$$

2.4 Struttura Simulink del modello matematico

A questo punto è importante riportare i blocchi Simulink del modello della dinamica del velivolo più rappresentativi al fine di capire meglio la strategia di programmazione utilizzata.

La schermata principale (vedi Figura 2.2) del *file* relativo a tale modello, denominato “*INS_non_lineare.mdl*”, visualizza i canali relativi al moto del baricentro (“Blocco Accelerometri”) e alla dinamica di assetto (“Blocco Giroscopi”):

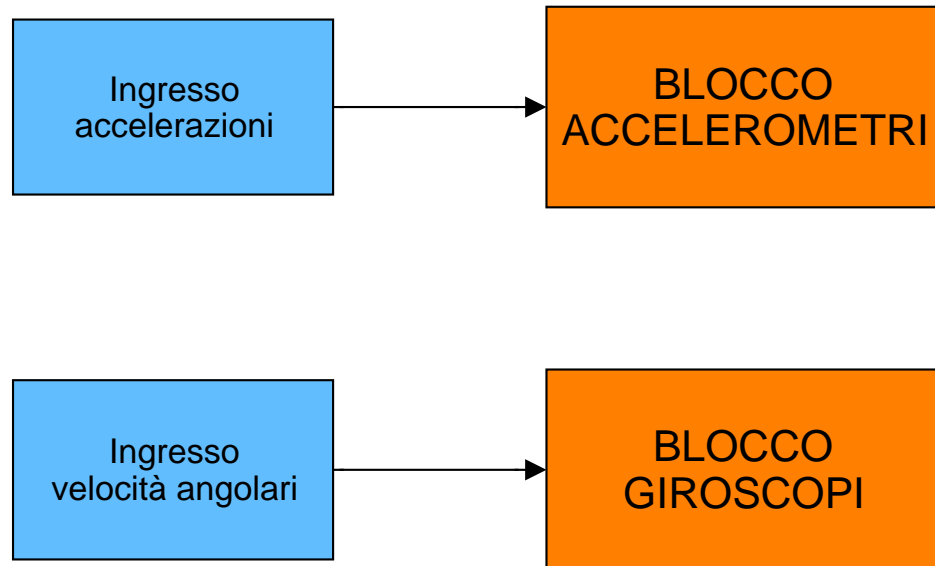


Figura 2.2: Blocco principale modello non lineare

2.4.1 Struttura Blocco Accelerometri

Il blocco “*Ingressi accelerazioni*” presenta tre canali relativi alle uscite dagli accelerometri della piattaforma inerziale “*simulata*”. Nella Figura 2.3, riportata come esempio, i segnali sono del tutto generici (sinusoidi) e non legati alla dinamica reale della piattaforma:

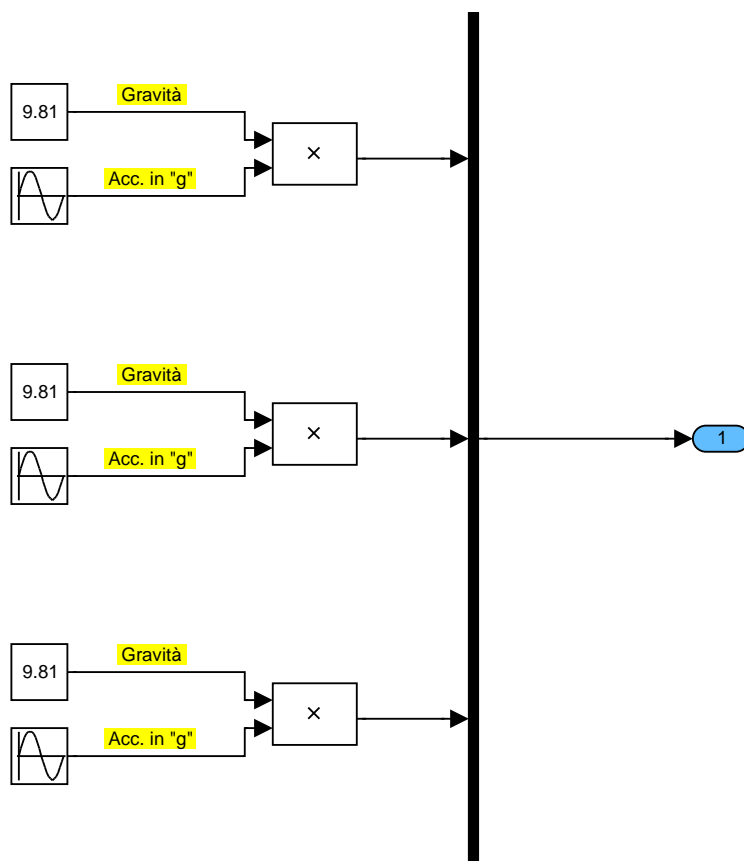


Figura 2.3: Blocco ingressi accelerazioni

Il “*Blocco Accelerometri*” contiene i sottoblocchi visualizzati in Figura 2.4 e come si intuisce dai loro nomi essi integrano all’interno le relative equazioni ricavate in §2.2 [vedi (2.6)÷(2.11)].

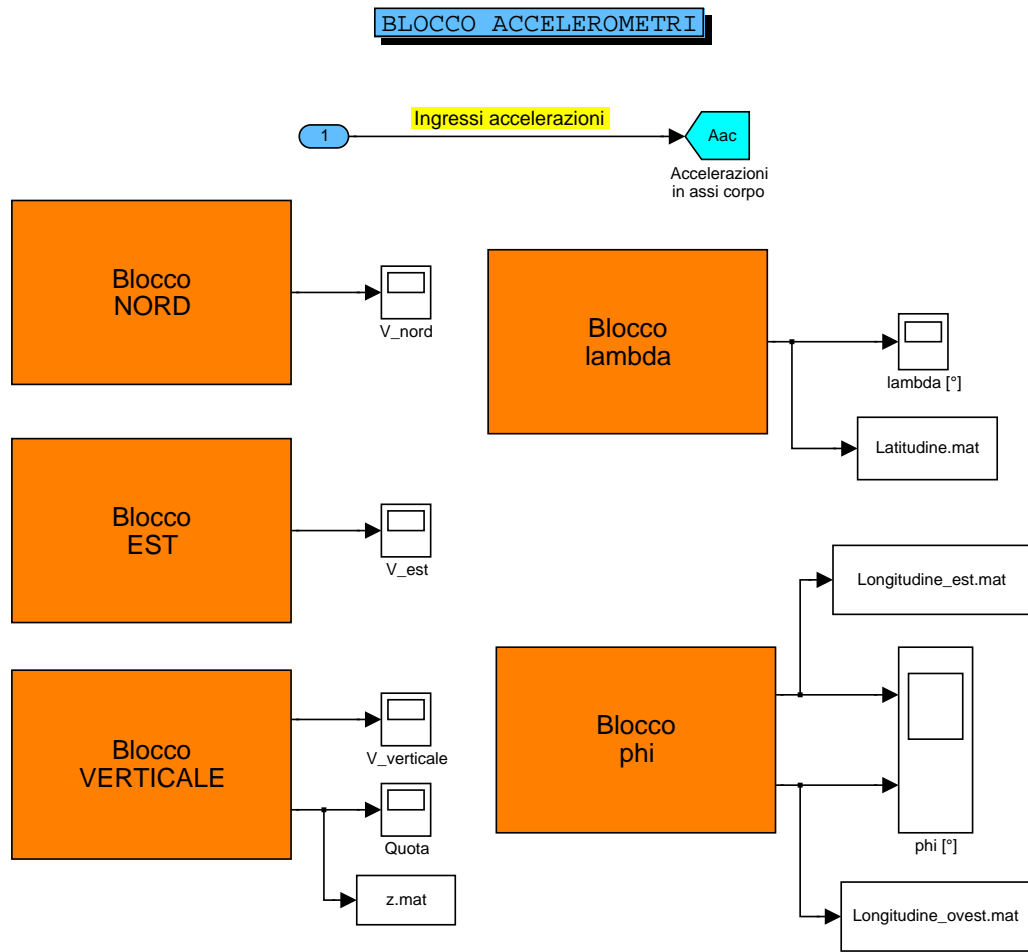


Figura 2.4: Blocco accelerometri

In particolare all'interno del blocco “*phi*” (vedi Figura 2.7) è stato inserito un ciclo “*if..else*” che permette di limitare l'escursione angolare in *longitudine* in un intervallo di $\pm 180^\circ$. Il blocco “*if*” in particolare interviene quando l'aereo ha una componente di velocità rivolta verso Ovest e quindi si avvicina al passaggio $+180^\circ \Rightarrow -180^\circ$ mentre il blocco “*else*” interviene nel caso opposto, cioè nel passaggio $-180^\circ \Rightarrow +180^\circ$:

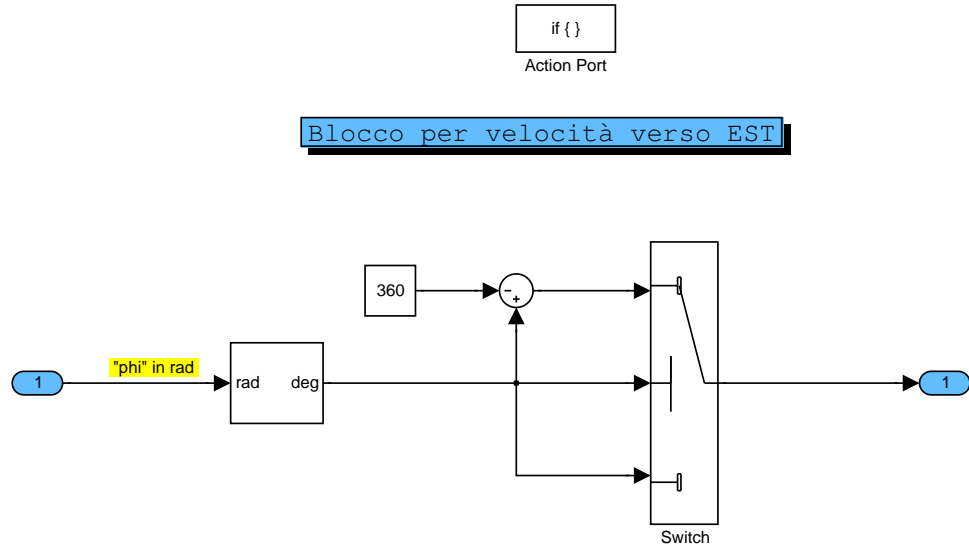


Figura 2.5: Limitazione longitudine a $+180^\circ$

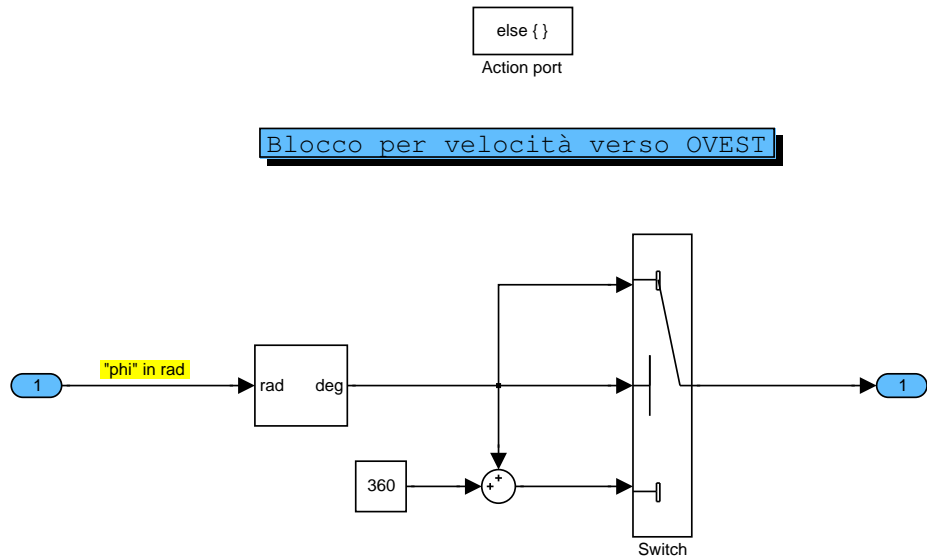


Figura 2.6: Limitazione longitudine a -180°

Per la *latitudine* tale controllo avviene naturalmente in quanto, dipendendo essa da V_{Nord} , nel momento in cui arrivo a latitudini pari a 90° tale velocità inverte di segno e l'angolo ricomincia a diminuire.

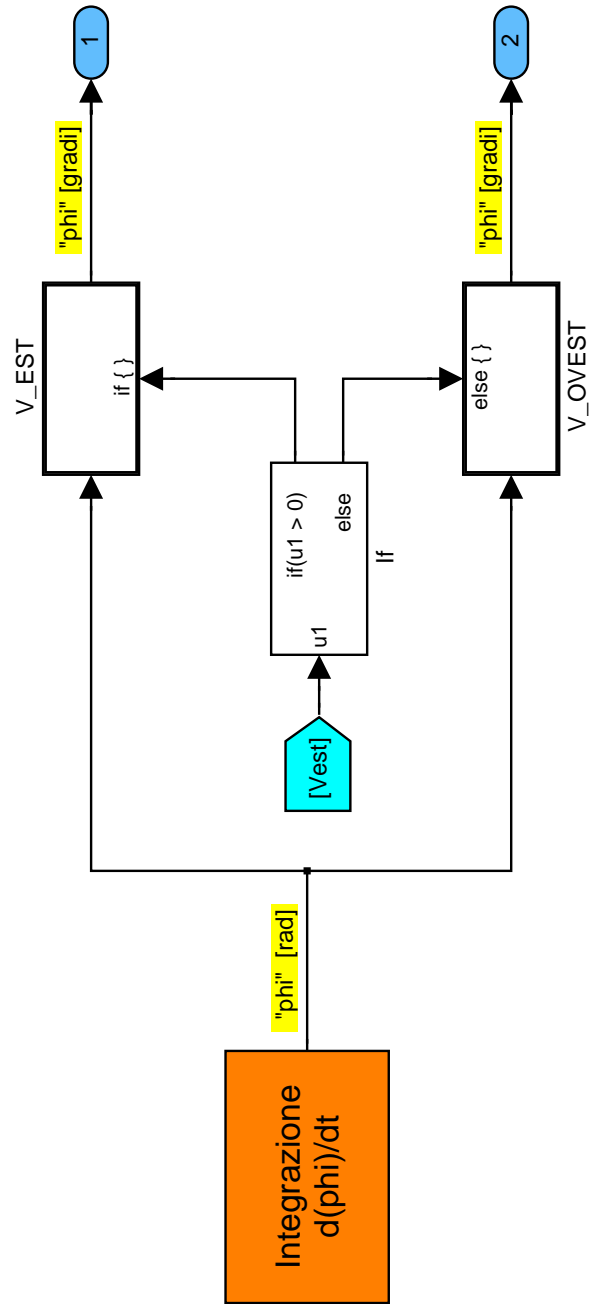


Figura 2.7: Blocco phi

A titolo di esempio si riporta la struttura del blocco “NORD” (vedi Figura 2.8), essendo le altre del tutto analoghe; in particolare si notano due blocchi funzione $[f(u)]$ nei quali viene scritta l’equazione (2.6) (il termine relativo agli ingressi in “ A_x to A_N ed il resto in “Forze apparenti”).

2.4.2 Struttura Blocco Giroscopi

Analogamente a quanto fatto per gli accelerometri riportiamo qui degli ingressi generici relativi alle velocità angolari:

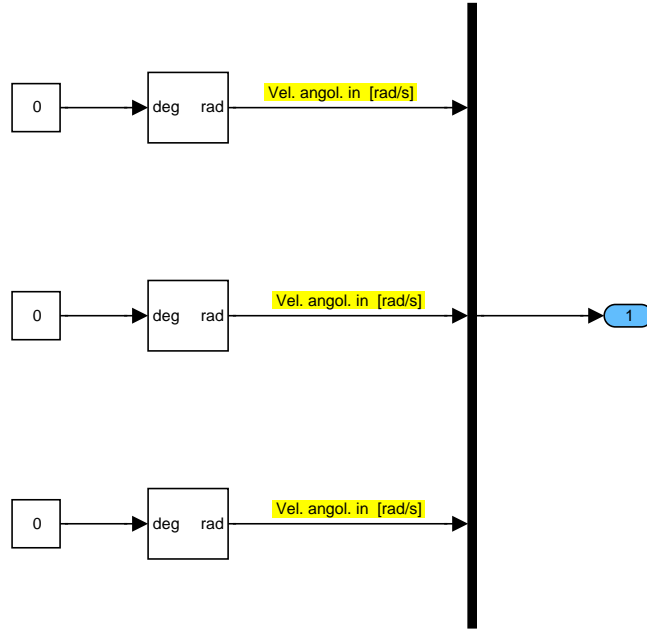


Figura 2.9: Blocco ingressi velocità angolari

Nella Figura 2.10 compaiono tre sottosistemi principali all'interno dei quali vengono implementate le equazioni viste in §2.3.2.

In particolare nel blocco “*Calcolo dei parametri di Eulero*” si calcolano le derivate relative alle equazioni differenziali (2.13)÷(2.16); in “*Calcolo degli angoli di Eulero*” si risolvono le equazioni (2.25)÷(2.27); in “*Condizioni iniziali*” si valutano le (2.20)÷(2.23).

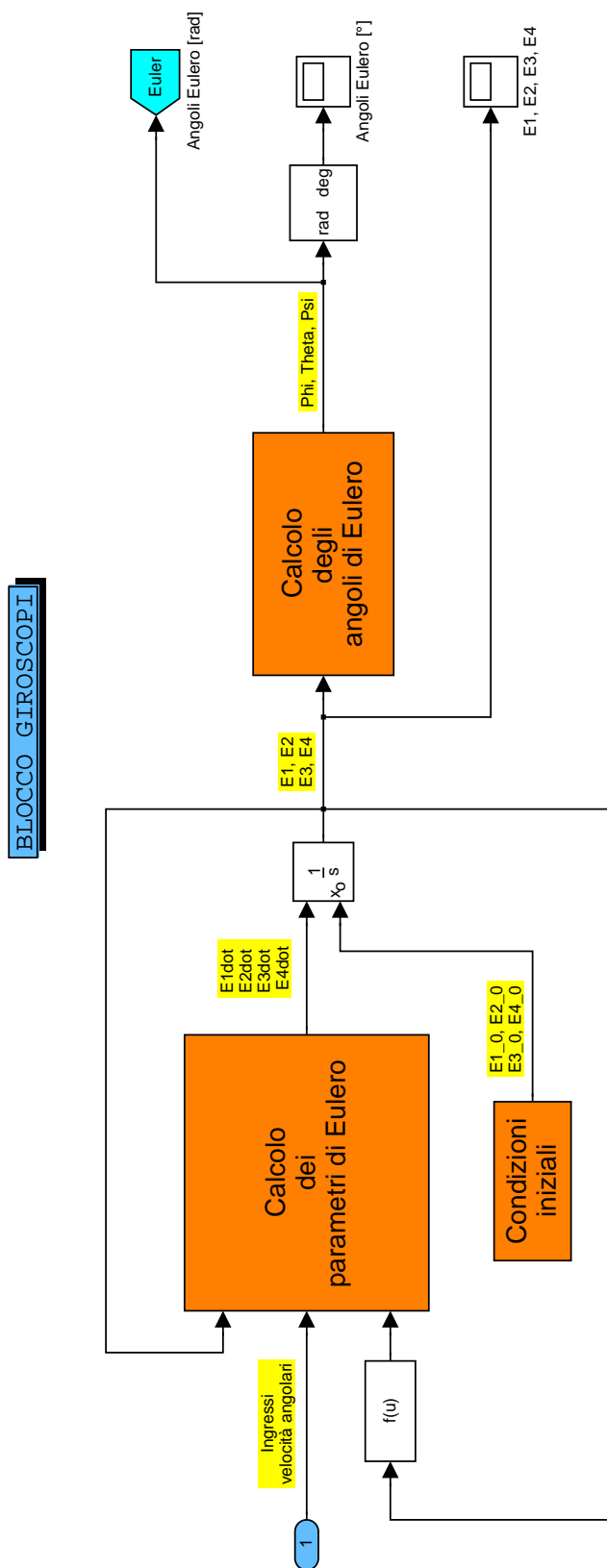


Figura 2.10: Blocco giroscopi

2.5 Esempio di simulazione

Al fine di validare il modello non lineare descritto in questo capitolo eseguiamo ora una simulazione avente le seguenti condizioni iniziali in ingresso:

$$\begin{aligned}
 T_{sim} &= 50000 \quad [s] \\
 V_N &= 100 \quad [m/s] \\
 V_E &= -50 \quad [m/s] \\
 V_D &= 0 \quad [m/s] \\
 \lambda &= 0 \quad [^\circ] \\
 \varphi &= 0 \quad [^\circ] \\
 H &= 5000 \quad [m] \\
 A_N &= 0 \quad [m/s^2] \\
 A_E &= 0 \quad [m/s^2] \\
 A_D &= 0 \quad [m/s^2] \\
 p &= 0 \quad [^\circ/s] \\
 q &= 0 \quad [^\circ/s] \\
 r &= 0 \quad [^\circ/s]
 \end{aligned}$$

Tali dati vengono inseriti in un *file Matlab* chiamato “*Inizializzazione.m*” che, una volta lanciato, li carica nel *workspace*; a questo punto è possibile avviare la simulazione vera e propria ¹.

I risultati ottenuti sono stati riportati su di una cartina geografica (utilizzando la proiezione cilindrica di Miller) al fine di rendere più comprensibili le variazioni di latitudine e longitudine sperimentate dal baricentro del velivolo durante la simulazione.

La Figura 2.11 rappresenta un’ingrandimento di tale traiettoria (curva gialla); si noti come in assenza di accelerazioni e velocità angolari dovute ad esempio a manovre eseguite da un pilota, il moto libero sia governato dalle sole forze legate alla rotazione terrestre (Coriolis in particolare):

¹NOTA : tale procedura è necessaria per tutte le simulazioni di cui si parlerà in seguito, ivi comprese quelle in *real time*

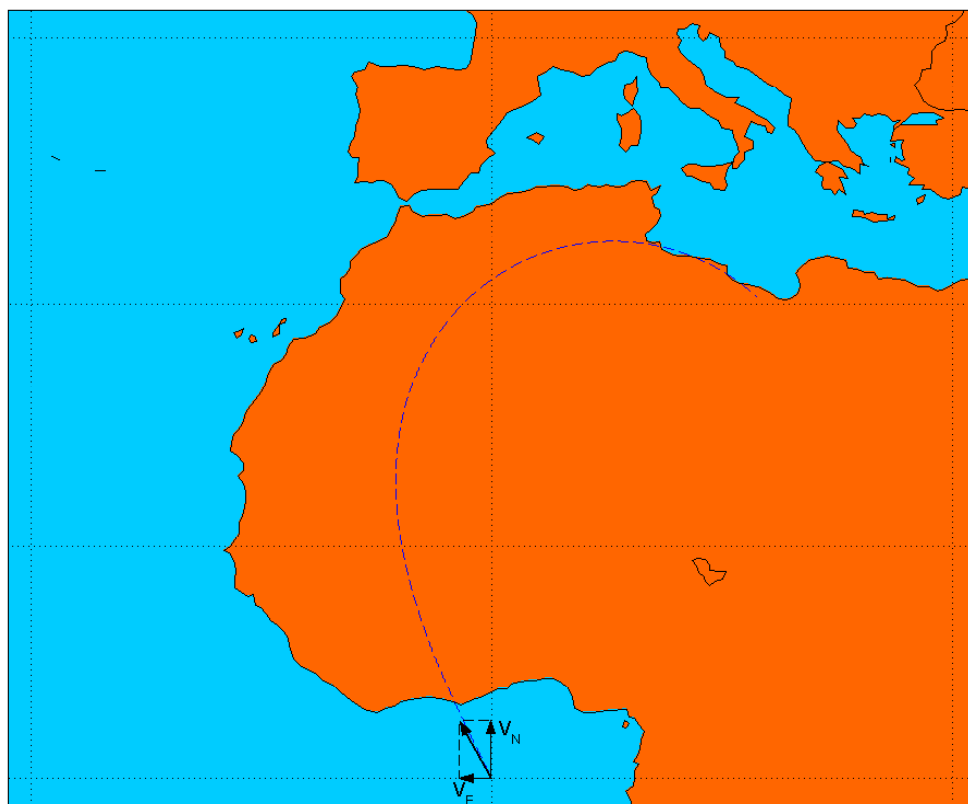


Figura 2.11: Simulazione con modello non lineare

3

Modello matematico linearizzato

3.1 Linearizzazione delle equazioni di moto e di assetto

Partendo dal sistema di equazioni differenziali non lineari viste in §2.2 e §2.3 [(2.6)÷(2.16)] e seguendo quanto indicato in appendice A si è ottenuto un sistema in variabili di stato del tipo:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (3.1)$$

in cui:

$$\mathbf{x} = [V_N \ V_E \ V_D \ \lambda \ \varphi \ H \ E_0 \ E_1 \ E_2 \ E_3]^T \quad (3.2)$$

$$\mathbf{u} = [A_x \ A_y \ A_z \ g \ p \ q \ r]^T \quad (3.3)$$

dove:

- A_x Accelerazione lungo asse rollio
- A_y Accelerazione lungo asse beccheggio
- A_z Accelerazione lungo asse imbardata

Le matrici jacobiane \mathbf{A} e \mathbf{B} , indicate in forma simbolica nelle (A.5) e

(A.6), diventano ¹:

$$\mathbf{A}(t) = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & 0 & a_{16} & a_{17} & a_{18} & a_{19} & a_{110} \\ a_{21} & a_{22} & a_{23} & a_{24} & 0 & a_{26} & a_{27} & a_{28} & a_{29} & a_{210} \\ a_{31} & a_{32} & 0 & a_{34} & 0 & a_{36} & a_{37} & a_{38} & a_{39} & a_{310} \\ a_{41} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a_{52} & 0 & a_{54} & 0 & a_{56} & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{77} & a_{78} & a_{79} & a_{710} \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{87} & a_{88} & a_{89} & a_{810} \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{97} & a_{98} & a_{99} & a_{910} \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{107} & a_{108} & a_{109} & a_{1010} \end{bmatrix} \quad (3.4)$$

$$\mathbf{B}(t) = \begin{bmatrix} b_{11} & b_{12} & b_{13} & 0 & 0 & 0 & 0 \\ b_{21} & b_{22} & b_{23} & 0 & 0 & 0 & 0 \\ b_{31} & b_{32} & b_{33} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & b_{75} & b_{76} & b_{77} \\ 0 & 0 & 0 & 0 & b_{85} & b_{86} & b_{87} \\ 0 & 0 & 0 & 0 & b_{95} & b_{96} & b_{97} \\ 0 & 0 & 0 & 0 & b_{105} & b_{106} & b_{107} \end{bmatrix} \quad (3.5)$$

dove:

¹nota : $a_{ij} = \partial f_i / \partial x_j$ e $b_{ik} = \partial f_i / \partial u_k$

$$\begin{aligned}
 a_{11} &= \frac{V_D}{R} & a_{12} &= -2(\omega_s \sin \lambda + \frac{V_E}{R} \tan \lambda) \\
 a_{13} &= \frac{V_N}{R} & a_{14} &= -V_E(2\omega_s \cos \lambda + \frac{V_E}{R \cos^2 \lambda}) \\
 a_{16} &= \frac{1}{R^2}(V_E^2 \tan \lambda - V_N V_V) & a_{17} &= 2E_0 A_x - 2E_3 A_y + 2E_2 A_z \\
 a_{18} &= 2E_1 A_x + 2E_2 A_y + 2E_3 A_z & a_{19} &= -2E_2 A_x + 2E_1 A_y + 2E_0 A_z \\
 a_{110} &= -2E_3 A_x - 2E_0 A_y + 2E_1 A_z & a_{21} &= 2\omega_s \sin \lambda + \frac{V_E}{R} \tan \lambda \\
 a_{22} &= \frac{1}{R}(V_N \tan \lambda + V_D) & a_{23} &= 2\omega_s \cos \lambda + \frac{V_E}{R} \\
 a_{24} &= 2\omega_s V_N \cos \lambda + \frac{V_N V_E}{R \cos^2 \lambda} - 2\omega_s V_D \sin \lambda & a_{26} &= -\frac{V_E}{R^2}(V_N \tan \lambda + V_D) \\
 a_{27} &= 2E_3 A_x + 2E_0 A_y - 2E_1 A_z & a_{28} &= 2E_2 A_x - 2E_1 A_y - 2E_0 A_z \\
 a_{29} &= 2E_1 A_x + 2E_2 A_y + 2E_3 A_z & a_{210} &= 2E_0 A_x - 2E_3 A_y + 2E_2 A_z \\
 a_{31} &= \frac{-2V_N}{R} & a_{32} &= -2(\omega_s \cos \lambda + \frac{V_E}{R}) \\
 a_{34} &= 2\omega_s V_E \sin \lambda & a_{36} &= \frac{V_N^2 + V_E^2}{R^2} \\
 a_{37} &= -2E_2 A_x + 2E_1 A_y + 2E_0 A_z & a_{38} &= 2E_3 A_x + 2E_0 A_y - 2E_1 A_z \\
 a_{39} &= -2E_0 A_x + 2E_3 A_y - 2E_2 A_z & a_{310} &= 2E_1 A_x + 2E_2 A_y + 2E_3 A_z \\
 a_{41} &= \frac{1}{R} & a_{46} &= \frac{V_N}{R^2} \\
 a_{52} &= \frac{1}{R \cos \lambda} & a_{54} &= \frac{V_E \tan \lambda}{R \cos \lambda} \\
 a_{56} &= -\frac{V_E}{R^2 \cos \lambda} & a_{77} &= -2E_0^2 \\
 a_{78} &= -\frac{p}{2} + 2E_0 E_1 & a_{79} &= -\frac{q}{2} - 2E_0 E_2 \\
 a_{710} &= -\frac{r}{2} - 2E_0 E_3 & a_{87} &= \frac{p}{2} - 2E_0 E_1 \\
 a_{88} &= -2E_1^2 & a_{89} &= \frac{r}{2} - 2E_1 E_2 \\
 a_{810} &= -\frac{q}{2} - 2E_1 E_3 & a_{97} &= \frac{q}{2} - 2E_0 E_2 \\
 a_{98} &= -\frac{r}{2} - 2E_1 E_2 & a_{99} &= -2E_2^2 \\
 a_{910} &= \frac{p}{2} - 2E_2 E_3 & a_{107} &= \frac{r}{2} - 2E_0 E_3 \\
 a_{108} &= \frac{q}{2} - 2E_1 E_3 & a_{109} &= -\frac{p}{2} - 2E_2 E_3 \\
 a_{1010} &= -2E_3^2
 \end{aligned}$$

$$b_{11} = E_0^2 + E_1^2 - E_2^2 - E_3^2$$

$$b_{12} = 2(E_1 E_2 - E_0 E_3)$$

$$b_{13} = 2(E_1 E_3 + E_0 E_2)$$

$$b_{21} = 2(E_1 E_2 + E_0 E_3)$$

$$b_{22} = E_0^2 - E_1^2 + E_2^2 - E_3^2$$

$$b_{23} = 2(E_2 E_3 - E_0 E_1)$$

$$b_{31} = 2(E_1 E_3 - E_0 E_2)$$

$$b_{32} = 2(E_2 E_3 + E_0 E_1)$$

$$b_{33} = E_0^2 - E_1^2 - E_2^2 + E_3^2$$

$$b_{75} = -\frac{E_1}{2}$$

$$b_{76} = -\frac{E_2}{2}$$

$$b_{77} = -\frac{E_3}{2}$$

$$b_{85} = \frac{E_0}{2}$$

$$b_{86} = -\frac{E_3}{2}$$

$$b_{87} = \frac{E_2}{2}$$

$$b_{95} = \frac{E_3}{2}$$

$$b_{96} = \frac{E_0}{2}$$

$$b_{97} = -\frac{E_1}{2}$$

$$b_{105} = -\frac{E_2}{2}$$

$$b_{106} = \frac{E_1}{2}$$

$$b_{107} = \frac{E_0}{2}$$

3.2 Struttura Simulink del modello matematico

In maniera del tutto analoga al modello non lineare analizzato nel precedente capitolo, si illustrano ora i blocchi Simulink più significativi.

Il blocco principale del *file* denominato “*INS_linearizzato.mdl*” (vedi Figura 3.1) si presenta con una struttura suddivisa in sottoblocchi rappresentanti le dieci variabili di stato (NORD, EST, DOWN, ecc.), gli ingressi e quattro vettori (“X”, “dX”, “d_omega”, “d_acc”):

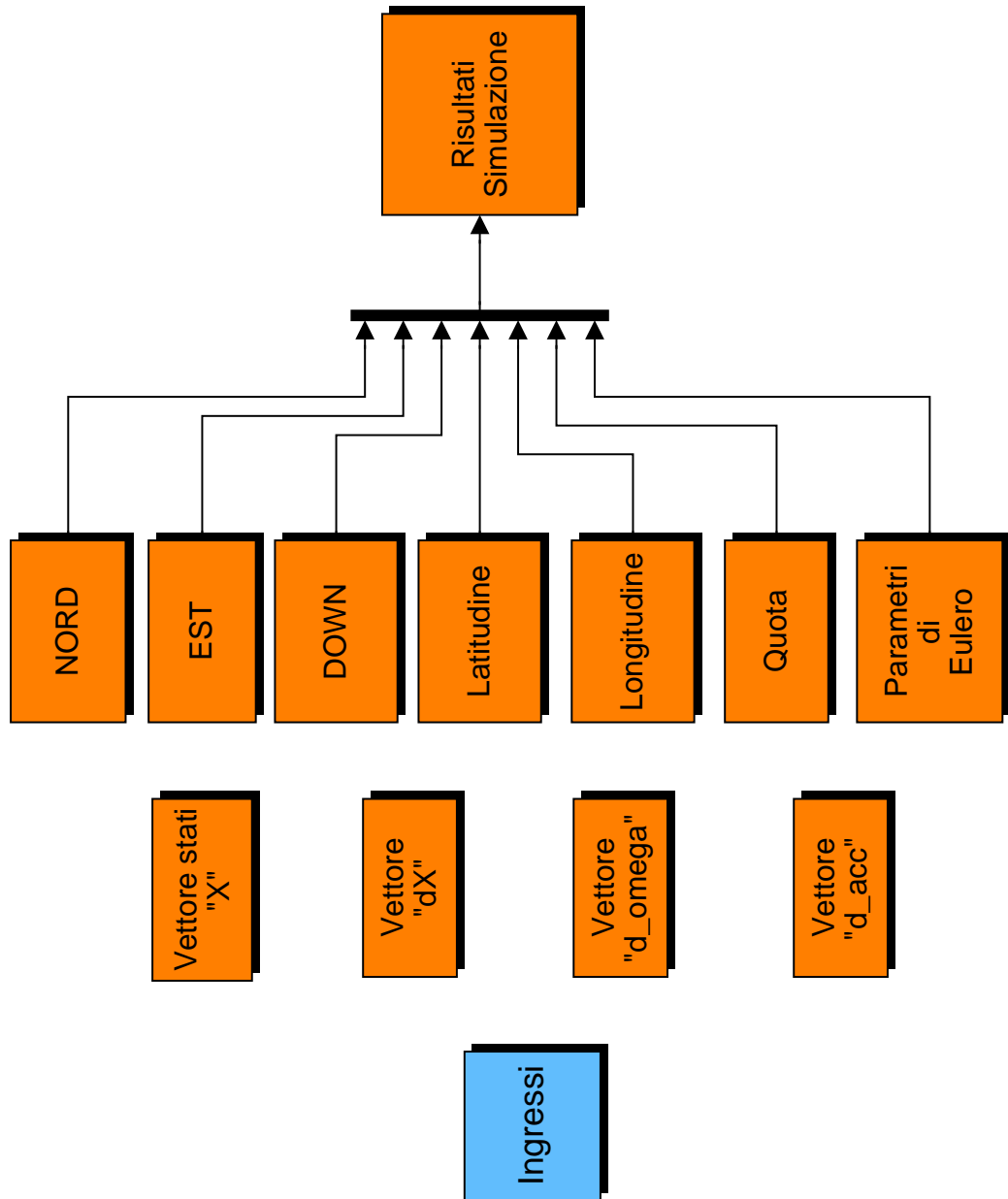


Figura 3.1: Blocco principale modello linearizzato

Questi ultimi in particolare sono una conseguenza della linearizzazione e rappresentano gli stati (Figura 3.2), i loro incrementi attorno alla condizione di “equilibrio” (Figura 3.3) e gli incrementi degli ingressi (Figure 3.4 e 3.5):

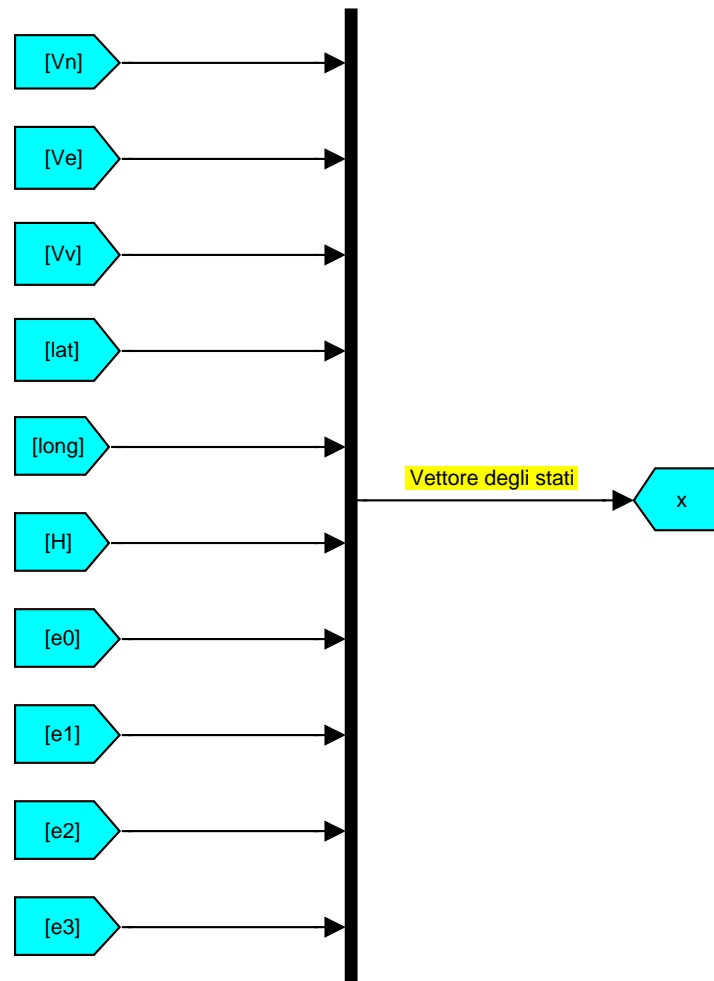


Figura 3.2: Blocco vettore degli stati

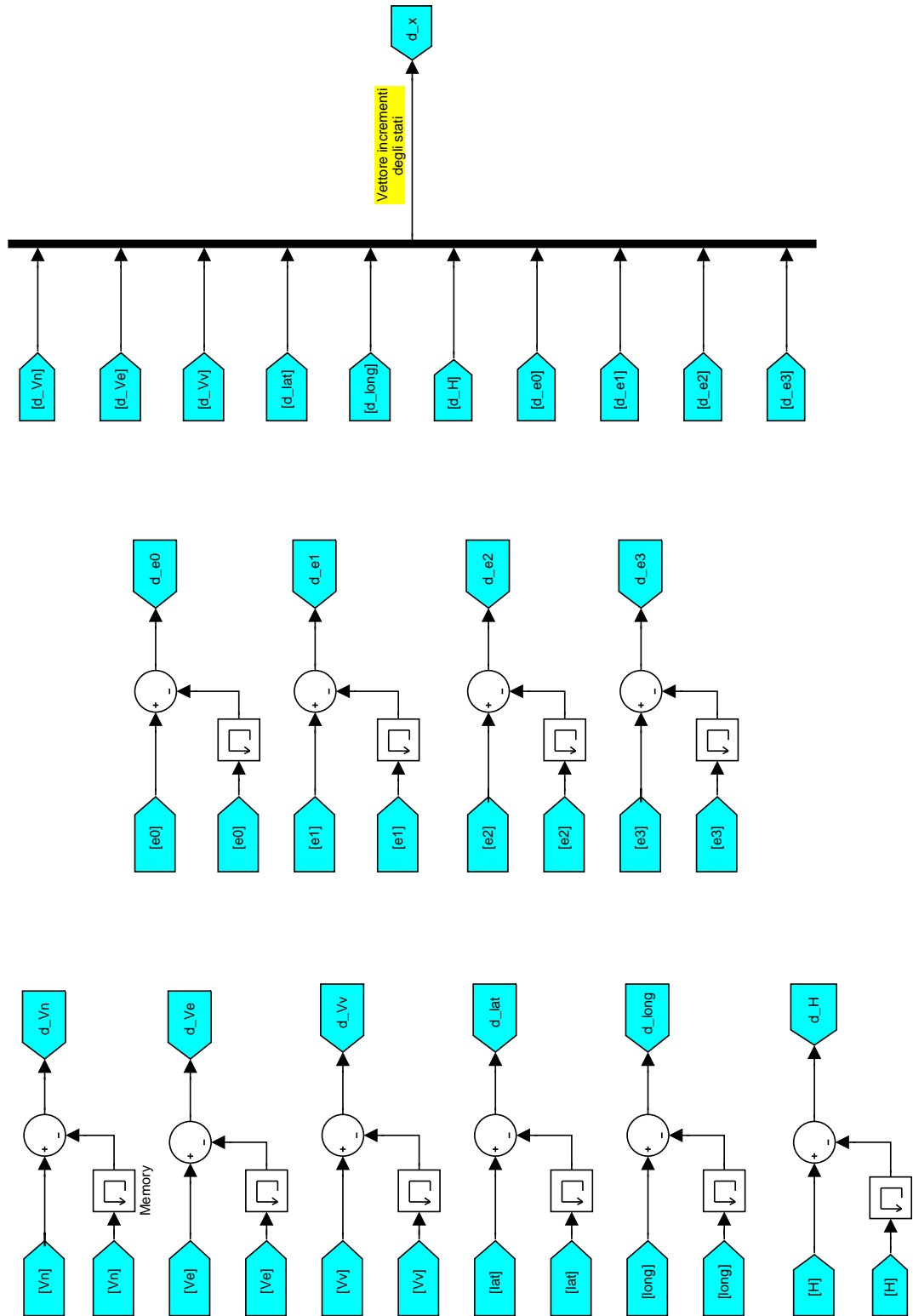


Figura 3.3: Blocco vettore incrementi degli stati

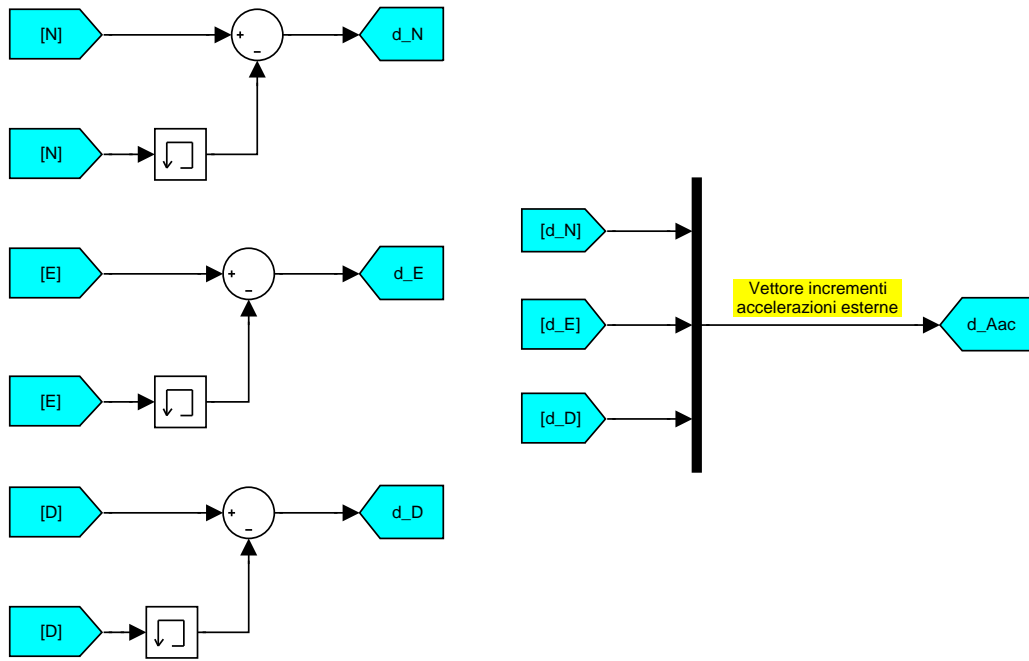


Figura 3.4: Blocco vettore incrementi delle velocità angolari

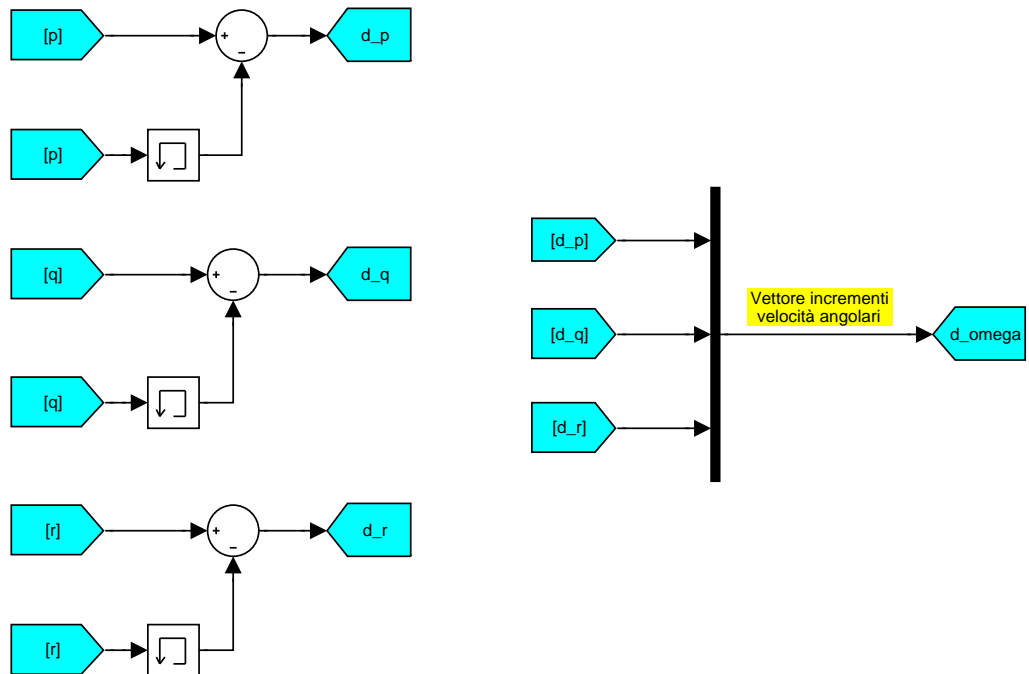


Figura 3.5: Blocco vettore incrementi delle accelerazioni

Nelle Figure 3.4 e 3.5 gli incrementi di accelerazione e velocità angolare (d_N , d_E , d_D , ecc.), sono frutto della differenza fra gli ingressi (N, E, D, ecc. vedi Figure 3.6 e 3.7) tra due passi successivi (mediante la funzione “Memory”):

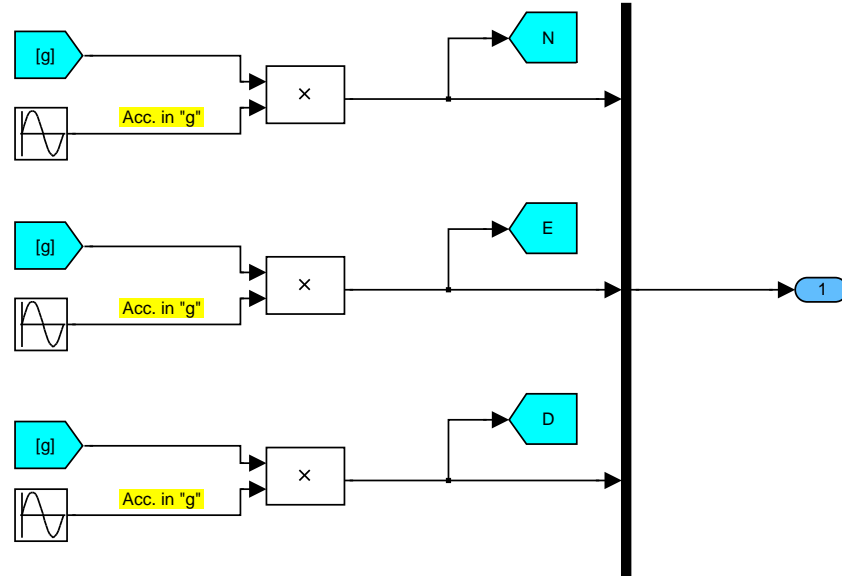


Figura 3.6: Blocco ingressi accelerazioni

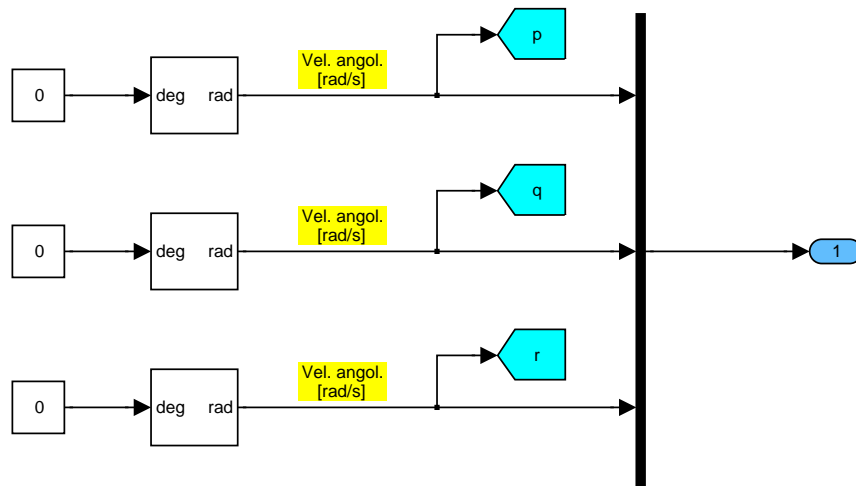


Figura 3.7: Blocco ingressi velocità angolari

Le equazioni relative ai vari stati sono state implementate all'interno dei blocchi di Figura 3.1 con forma molto simile tra loro. Pertanto si riportano due esempi rappresentativi:

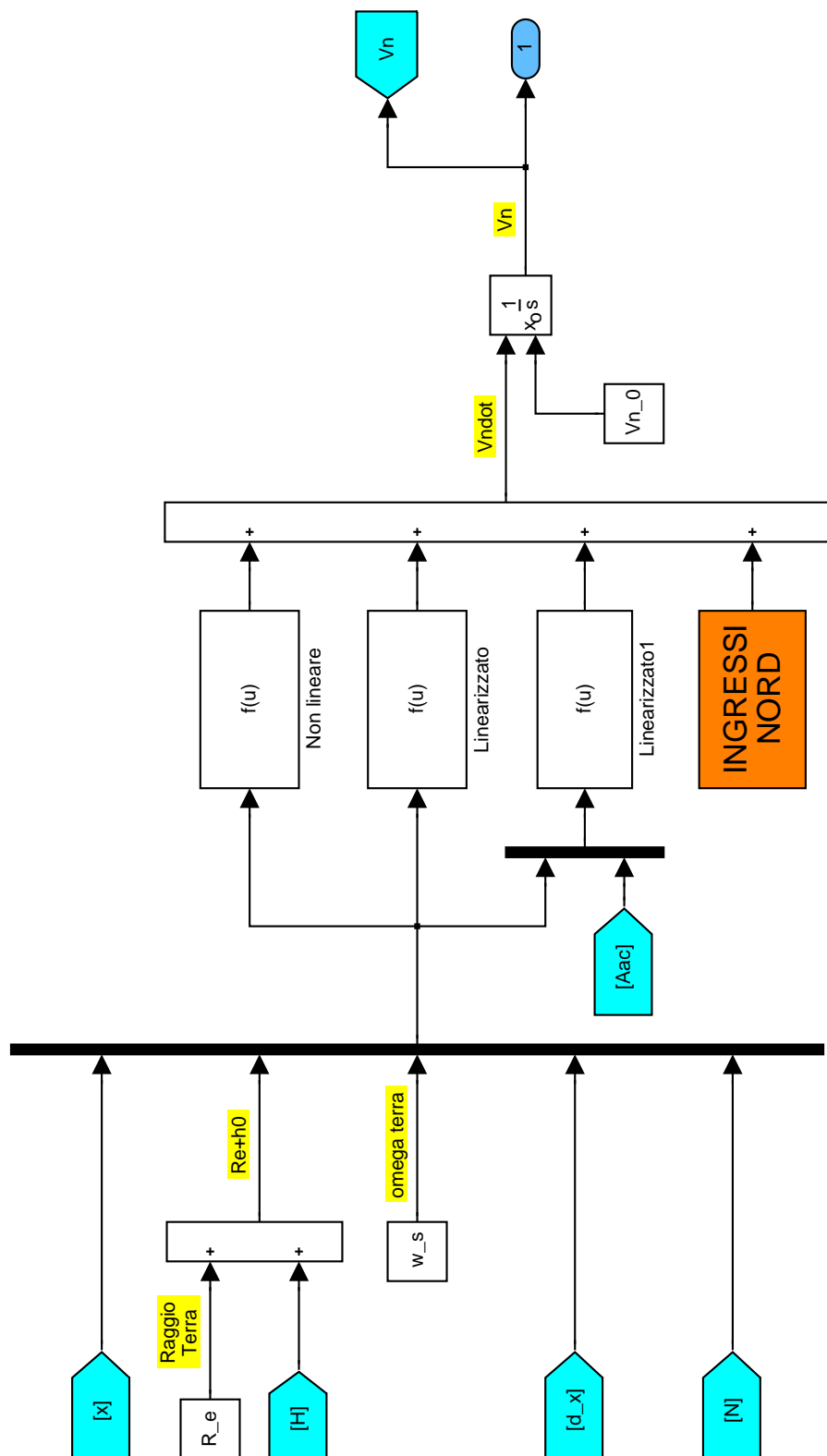


Figura 3.8: Blocco Nord

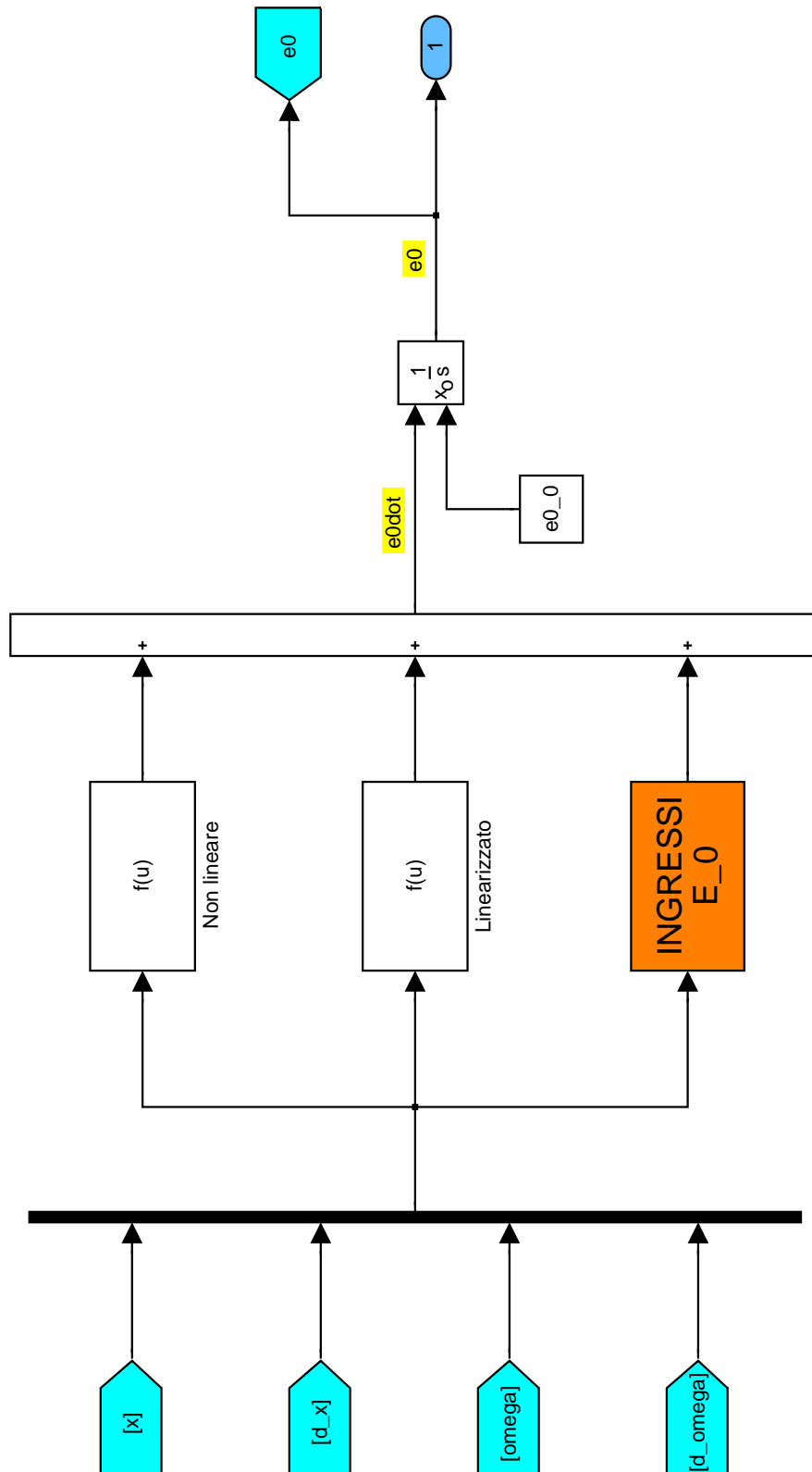


Figura 3.9: Blocco Parametri di Eulero (E_0)

Nelle Figure 3.8 e 3.9 si nota come i calcoli di \dot{V}_N (“Vndot”) e di \dot{E}_0 (“e0dot”) siano il risultato della somma di più componenti; nella prima ad esempio il blocco “Non lineare” calcola la condizione di “equilibrio” (punto di linearizzazione) mediante la relativa equazione non lineare (2.6), mentre i blocchi “Linearizzato”, “Linearizzato1” e “INGRESSI NORD” calcolano la prima riga dei prodotti $\mathbf{A}\mathbf{x}$ e $\mathbf{B}\mathbf{u}$ rispettivamente (vedi eq. (3.1)).

3.3 Esempio di simulazione

Per garantire l'accuratezza del modello linearizzato si procede ora con un esempio di simulazione, ponendo a confronto i risultati di tre modelli particolari che sono:

Modello “non lineare” i cui risultati sono presi a riferimento per gli altri due casi;

Modello “linearizzato non tempo-variante” nel quale la linearizzazione avviene unicamente al tempo zero (cioè le matrici jacobiane (3.4) e (3.5) vengono calcolate in quel momento e rimangono costanti per il resto della simulazione);

Modello “linearizzato tempo-variante” nel quale le matrici jacobiane di cui sopra vengono aggiornate ad ogni passo di simulazione.

Le condizioni iniziali comuni ai tre casi sono le seguenti:

$$\begin{aligned}
 T_{sim} &= 50000 \quad [s] \\
 V_N &= 200 \quad [m/s] \\
 V_E &= 0 \quad [m/s] \\
 V_D &= 0 \quad [m/s] \\
 \lambda &= 0 \quad [^\circ] \\
 \varphi &= 0 \quad [^\circ] \\
 H &= 5000 \quad [m] \\
 A_N &= 0 \quad [m/s^2] \\
 A_E &= 0 \quad [m/s^2] \\
 A_D &= 0 \quad [m/s^2] \\
 p &= 0 \quad [^\circ/s] \\
 q &= 0 \quad [^\circ/s] \\
 r &= 0 \quad [^\circ/s]
 \end{aligned}$$

Le tracce a terra relative ai tre casi sono riportate in Figura 3.10 rispettivamente nei colori blu, verde e giallo. In particolare si noti dalla Figura 3.11 come la *curva verde* tenda a scostarsi dal comportamento reale (evidenziato dalla *linea blu*) dopo circa 2000 secondi, mentre quella *gialla* vi sia

praticamente sovrapposta. Alla luce di questi risultati si è ovviamente scelto di utilizzare nel seguito di questo lavoro il modello “*tempo variante*”, dato che i tempi di missione di un comune volo di ricognizione di un UAV sono dell’ordine delle dodici ore.

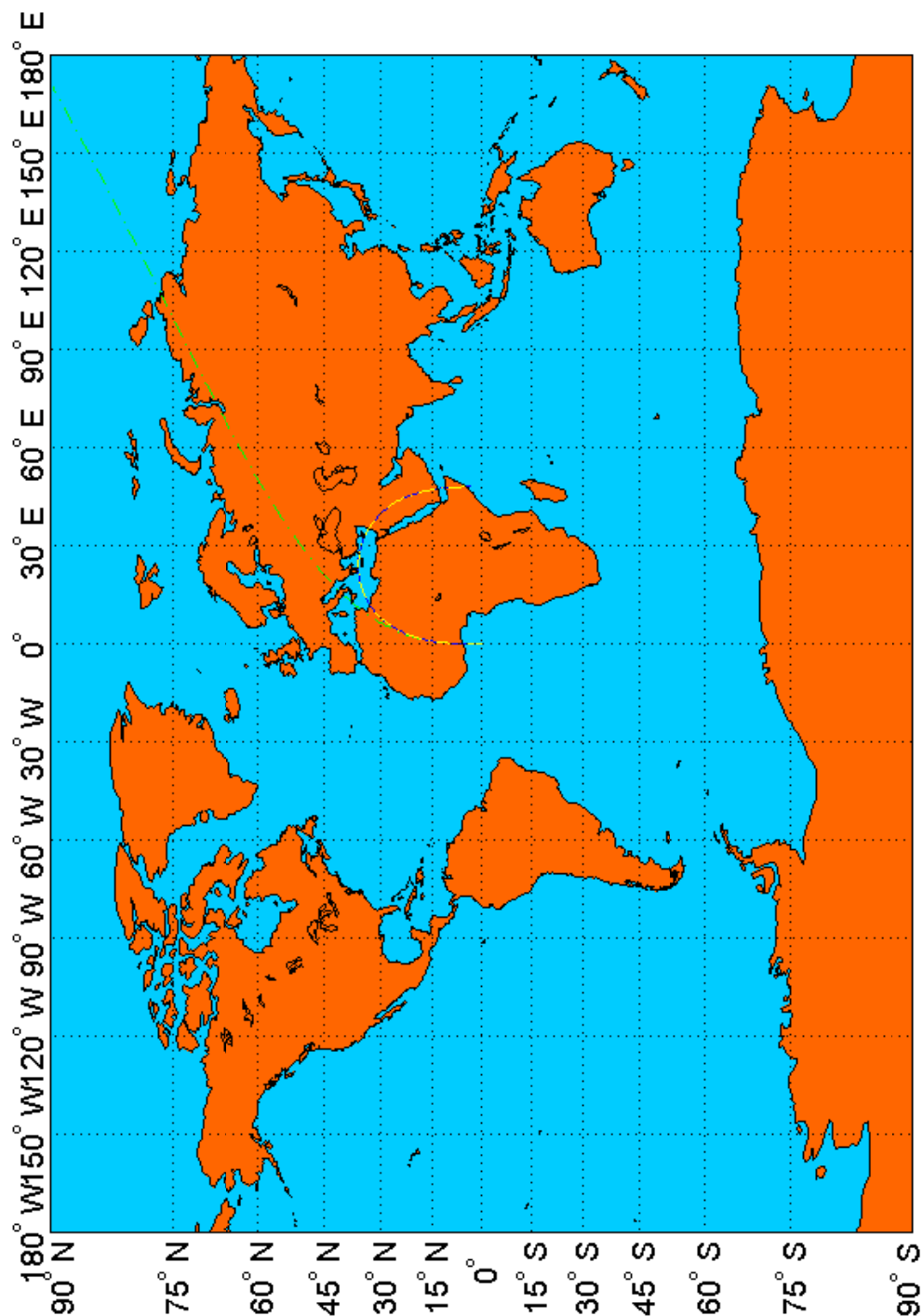


Figura 3.10: Esempi di simulazione a confronto

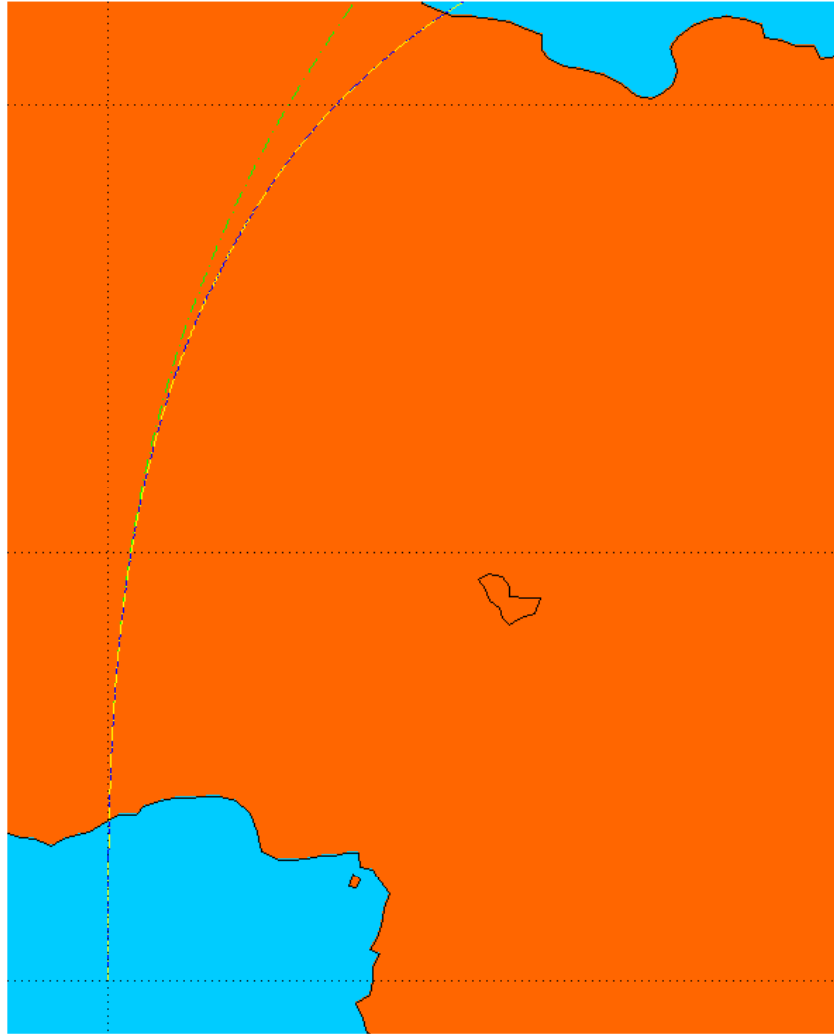


Figura 3.11: Dettaglio della zona di scostamento

4

Filtro di Kalman

4.1 Generalità

Il *filtro di Kalman* è un algoritmo di elaborazione ricorsiva di dati affetti da rumore, finalizzato ad ottimizzare la valutazione di un segnale in uscita. Esso utilizza una o più sorgenti di misura (“osservazioni”) per stimare lo stato corrente delle variabili di interesse.

Gli “ingredienti” base di tale strumento matematico sono:

1. La conoscenza di un modello dinamico del fenomeno fisico in esame
2. La descrizione statistica dei rumori, degli errori delle misurazioni e delle incertezze sui modelli
3. Le condizioni iniziali delle variabili di interesse

Ad esempio, per correggere la velocità ricavata per integrazione dal sistema inerziale di un velivolo si possono usare come sorgente di misura un apparato Doppler radar, un tubo di Pitot, un GPS o le informazioni uscenti dall’Air Data System; la cosa essenziale è che le quantità messe a confronto siano della stessa natura (velocità con velocità, posizioni con posizioni, ecc.). Per comprendere meglio qualitativamente come funziona il meccanismo di correzione, si riporta il seguente schema a blocchi:

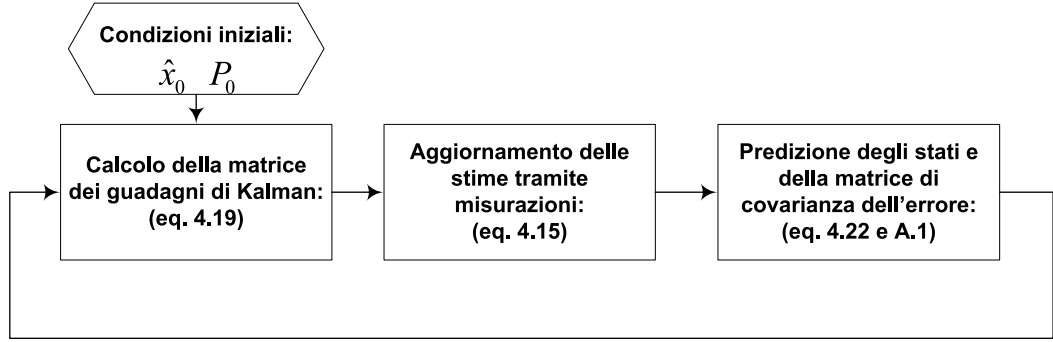


Figura 4.1: Schema algoritmo di calcolo filtro di Kalman

4.2 Richiami sui segnali aleatori

Per rendere più chiara la trattazione dei prossimi paragrafi, si richiamano qui brevemente alcune nozioni sui segnali *aleatori* e in particolare sui processi gaussiani bianchi.

4.2.1 Funzioni caratteristiche dei processi aleatori

Quando si parla di fenomeni aleatori, è importante ricordare cosa si intenda per media, valore quadratico medio, varianza, funzione di correlazione ecc. Riassumiamo perciò tali formule brevemente:

Media

$$\mu(t) = E[X(t)] = \int_{-\infty}^{+\infty} x f_X(x, t) dx \quad (4.1)$$

Valore quadratico medio

$$E[X^2(t)] = \int_{-\infty}^{+\infty} x^2 f_X(x, t) dx \quad (4.2)$$

Varianza

$$\sigma^2(t) = E[(X(t) - \mu(t))^2] = \int_{-\infty}^{+\infty} [x - \mu(t)]^2 f_X(x, t) dx \quad (4.3)$$

Funzione di correlazione

$$\rho(t_1, t_2) = E[X(t_1)X(t_2)] = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x_1 x_2 f_{X_1 X_2}(x_1, x_2; t_1, t_2) dx_1 dx_2 \quad (4.4)$$

dove

X	Variabile aleatoria in esame
$f_X(x, t)$	Funzione densità di probabilità associata alla variabile X nel dominio x
$E[...]$	Rappresenta l'operatore media in senso probabilistico

Nel nostro caso X è un vettore e tutte le definizioni viste in questo paragrafo si modificano di conseguenza. In particolare la funzione di correlazione si trasforma nella matrice omonima:

$$\mathbf{R}(t_1, t_2) = E[X(t_1)X^T(t_2)] = \begin{bmatrix} E[X_1(t_1)X_1(t_2)] & \cdots & E[X_1(t_1)X_n(t_2)] \\ \vdots & \ddots & \vdots \\ E[X_n(t_1)X_1(t_2)] & \cdots & E[X_n(t_1)X_n(t_2)] \end{bmatrix} \quad (4.5)$$

in cui gli elementi della diagonale vengono chiamati “*funzioni di autocorrelazione*”.

Nel presente lavoro si ipotizza che gli elementi del vettore X siano scorrelati tra loro; \mathbf{R} diventa perciò diagonale e prende il nome di “*matrice di covarianza*”.

4.2.2 Rumore “bianco”

Il rumore bianco è un'astrazione matematica con cui si indica un segnale aleatorio a media nulla e densità spettrale costante per tutte le frequenze. In formule questa definizione si traduce in:

$$\mu(t) = 0; \quad (4.6)$$

$$S(\omega) = \int_{-\infty}^{+\infty} \rho(t)e^{-j\omega t} dt = Q \quad \text{costante } \forall \omega \quad (4.7)$$

dove $S(\omega)$ è la *trasformata di Fourier* della funzione di correlazione vista in

(4.4). Nel caso in cui si abbia un vettore costituito da rumore bianco, la matrice di correlazione vista in (4.5) diventa:

$$\mathbf{R}(t) = \mathbf{Q}\delta(t) \quad (4.8)$$

dove \mathbf{Q} è una matrice quadrata e $\delta(t)$ è un impulso unitario nell'origine.

4.3 Modello matematico

Nei processi reali gli ingressi ad un sistema sono spesso affetti da rumore ed è per questo che non è possibile sempre usare gli osservatori convenzionali.

Def. : un osservatore è uno strumento matematico che permette di stimare le variabili di stato sulla base della conoscenza delle uscite e delle variabili di controllo [5].

Consideriamo il sistema lineare, non stazionario e continuo già trattato nel §3.1 a cui aggiungiamo del rumore bianco:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{v} \quad (4.9)$$

$$\mathbf{x}(0) = \mathbf{x}_0 \quad (4.10)$$

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{w} \quad (4.11)$$

dove la (4.11) rappresenta il legame fra le misure e gli stati mentre \mathbf{v} e \mathbf{w} sono i rumori di cui sopra, mutuamente scorrelati, aventi matrici di covarianza date da:

$$E[\mathbf{v}(t)\mathbf{v}^T(\tau)] = \mathbf{V}(t)\delta(t - \tau) \quad (4.12)$$

$$E[\mathbf{w}(t)\mathbf{w}^T(\tau)] = \mathbf{W}(t)\delta(t - \tau) \quad (4.13)$$

Le matrici \mathbf{A} e \mathbf{B} sono le stesse viste nel §3.1, mentre la matrice \mathbf{H} è lo jacobiano associato alle misure di λ , φ , H , V_{Nord} , V_{Est} e V_{Down} rispettivamente. Al fine di esplicitare per il caso in esame tale matrice, applichiamo la definizione data dalla (A.9), in cui le componenti del vettore $\mathbf{h}(\mathbf{x})$ hanno la seguente forma:

$$\begin{aligned} h_1 &= \lambda; & h_2 &= \varphi; & h_3 &= H; \\ h_4 &= V_{Nord}; & h_5 &= V_{Est}; & h_6 &= V_{Down}; \end{aligned} \quad (4.14)$$

Essa ha la seguente forma:

$$\mathbf{H}(t) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.15)$$

Consideriamo ora un osservatore per il modello (4.9) (vedi Figura 4.2):

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{K}(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}) \quad (4.16)$$

$$\hat{\mathbf{x}}(0) = \hat{\mathbf{x}}_0 \quad (4.17)$$

e si definisca l'errore di stima come

$$\mathbf{e}(t) = \mathbf{x}(t) - \hat{\mathbf{x}}(t) \quad (4.18)$$

avente valore quadratico medio

$$\mathbf{P} = E[\mathbf{e}^T(t)\mathbf{e}(t)] \quad (4.19)$$

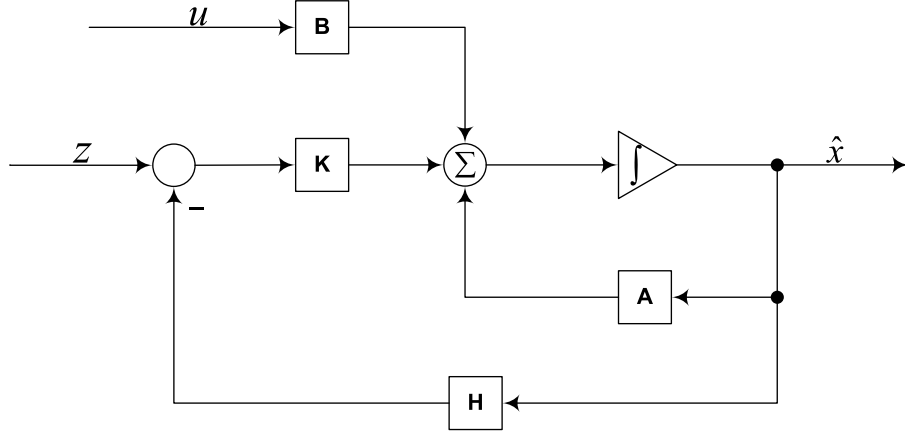


Figura 4.2: Schema dell'osservatore (filtro di Kalman)

Il progetto dell'*osservatore ottimo* consiste nello scegliere $\hat{\mathbf{x}}_0$ e \mathbf{K} in modo da minimizzare l'errore quadratico medio di stima (4.19). La soluzione di questo problema è stata trovata nel 1961 da *R.E. Kalman R.S. Bucy* nel modo seguente:

- Si sostituiscono le equazioni (4.9) e (4.16) in (4.18)
- Si ricava l'espressione di \mathbf{P} dalla (4.19)
- Si deriva quest'ultima rispetto a \mathbf{K} e si pone a zero il risultato ottenuto
- Si trova \mathbf{K} da questa equazione ottenendo il risultato seguente

$$\mathbf{K} = \mathbf{P}\mathbf{H}^T\mathbf{W}^{-1} \quad (4.20)$$

$$\hat{\mathbf{x}}_0 = \bar{\mathbf{x}}_0 \quad (4.21)$$

in cui $\bar{\mathbf{x}}_0$ è la media del vettore condizione iniziale degli stati (4.10)

$$E[\mathbf{x}_0] = \bar{\mathbf{x}}_0 \quad (4.22)$$

La matrice di covarianza dell'errore \mathbf{P} è soluzione dell'equazione di *Riccati* (4.23), qui riportata con la rispettiva condizione iniziale:

$$\dot{\mathbf{P}} = \mathbf{A}\mathbf{P} + \mathbf{P}\mathbf{A}^T - \mathbf{P}\mathbf{H}^T\mathbf{W}^{-1}\mathbf{H}\mathbf{P} + \mathbf{F}\mathbf{V}\mathbf{F}^T \quad (4.23)$$

$$\mathbf{P}(0) = \mathbf{P}_0 \quad (4.24)$$

con \mathbf{P}_0 ricavabile mediante

$$\mathbf{P}_0 = E[(\mathbf{x} - \bar{\mathbf{x}}_0)(\mathbf{x} - \bar{\mathbf{x}}_0)^T] \quad (4.25)$$

Concludendo, possiamo dire che il filtro di Kalman è uno strumento lineare che fornisce la stima degli stati con errore quadratico medio minimo; in altre parole non esiste nessun altro filtro lineare o non lineare che elaborando linearmente le misure \mathbf{z} e gli ingressi \mathbf{u} fornisca una stima con errore minore.

4.4 Struttura Simulink del filtro di Kalman

Il modello *Simulink* del filtro di Kalman, denominato “*LKF.mdl*”, è costituito da quattro fondamentali blocchi (vedi Figura 4.3):

Il blocco “*INS*” in cui vengono implementate le equazioni di navigazione mediante il modello “non lineare” visto nel capitolo 2;

Il blocco “*GPS*” contenente le misure necessarie alla correzione degli stati (posizione, quota e velocità);

Il blocco “*n-Hz*” nel quale i segnali delle misure (posizione e velocità) provenienti dal *GPS* vengono campionati alle frequenze desiderate;

Il blocco “*Kalman*” in cui si elabora il modello del filtro di Kalman visto nel §4.3 ed in particolare le equazioni (4.15), (4.16), (4.20) e (4.23).

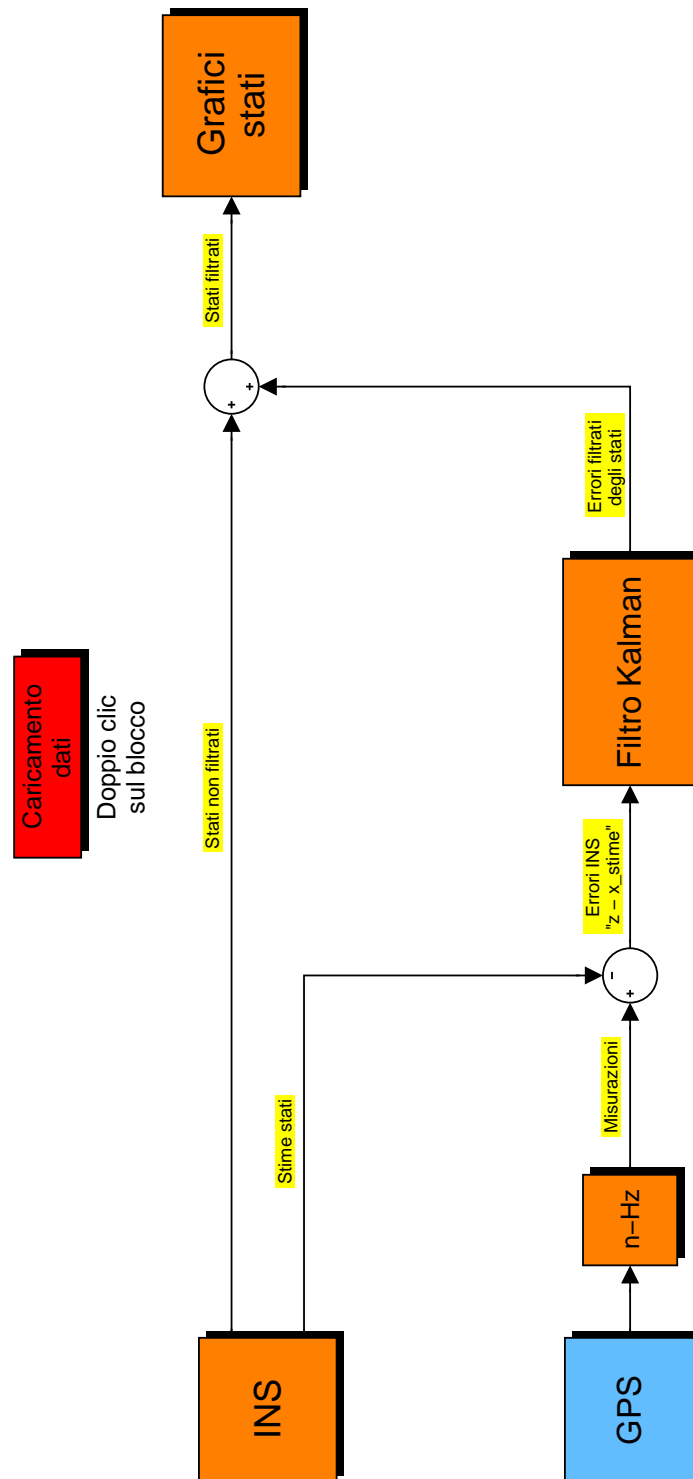


Figura 4.3: Blocco principale modello completo

4.4.1 Blocco *GPS*

In questa fase il ruolo del *GPS* è stato simulato mediante un modello identico a quello visto per il blocco *INS*. L'unica differenza sostanziale è che, mentre in quest'ultimo si integrano i segnali di ingresso di accelerazione e velocità angolari affetti da rumore (vedi Figura 4.4) generando perciò una deriva degli stati, nel *GPS* il rumore viene sommato a valle delle integrazioni, in perfetto accordo peraltro con la realtà.

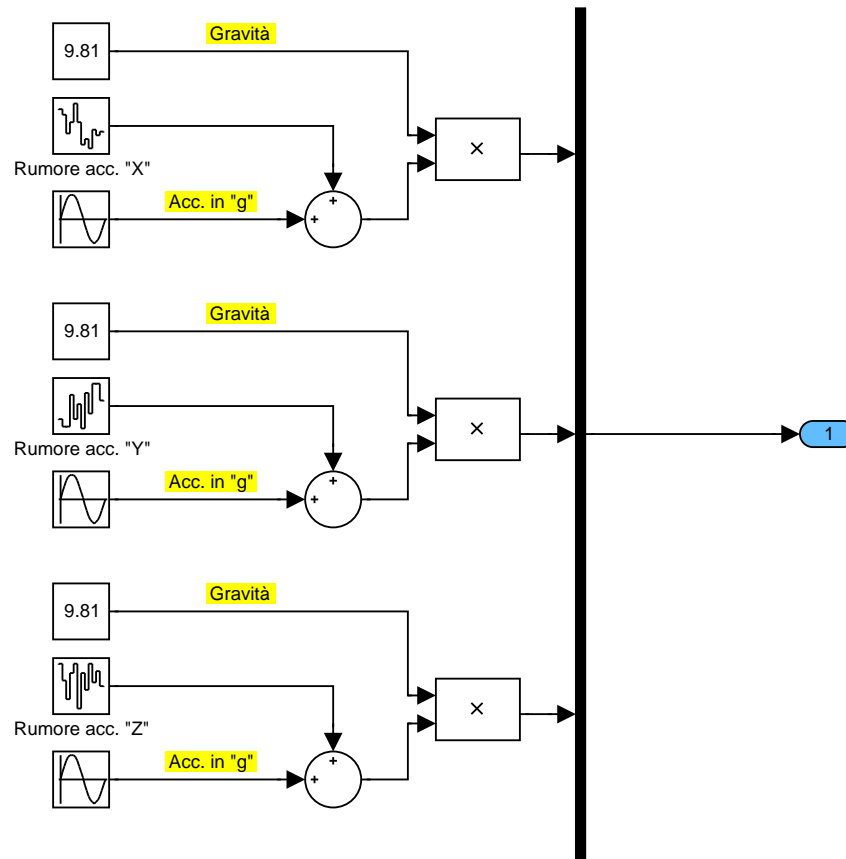


Figura 4.4: Ingressi accelerazioni INS con rumore

4.4.2 Blocco *n-Hz*

La Figura 4.5 mostra il contenuto del blocco di campionamento; la costante in rosso rappresenta il parametro di settaggio degli *Hertz* desiderati (nell'esempio riportato si campiona a 10Hz):



Nel blocco “*Matrici H e K*” (vedi Figura 4.7) si integra l’equazione differenziale (4.23) e nel contempo vengono calcolate le matrici H e K definite nell’ordine dalle (4.15) e (4.20). Tutte queste operazioni sono svolte da una *S-Function* scritta in linguaggio *C++* (vedi Appendice B) dal nome *Matrici_K_H.c*.

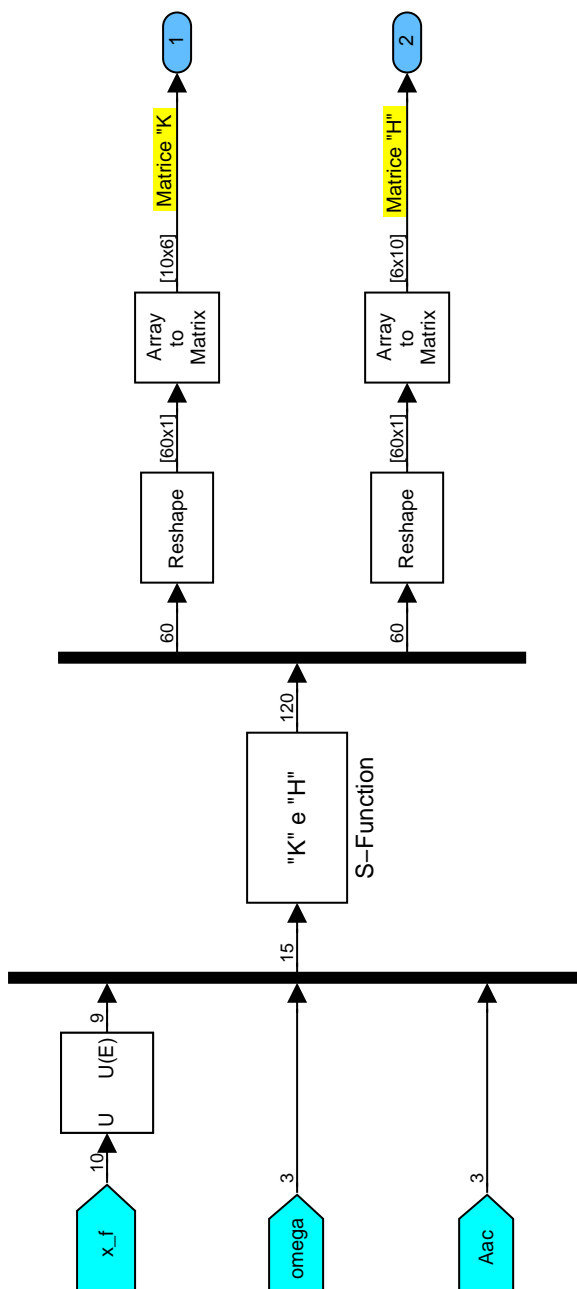


Figura 4.7: Blocco calcolo matrici “H” e “K”

Nel blocco denominato “*Integrazione equazioni del filtro*” viene implementato il modello linearizzato di Kalman definito dall’equazione (4.16); l’aspetto esteriore è identico a quello visto in Figura 3.1 mentre l’interno dei blocchi relativi agli stati ha la seguente struttura:

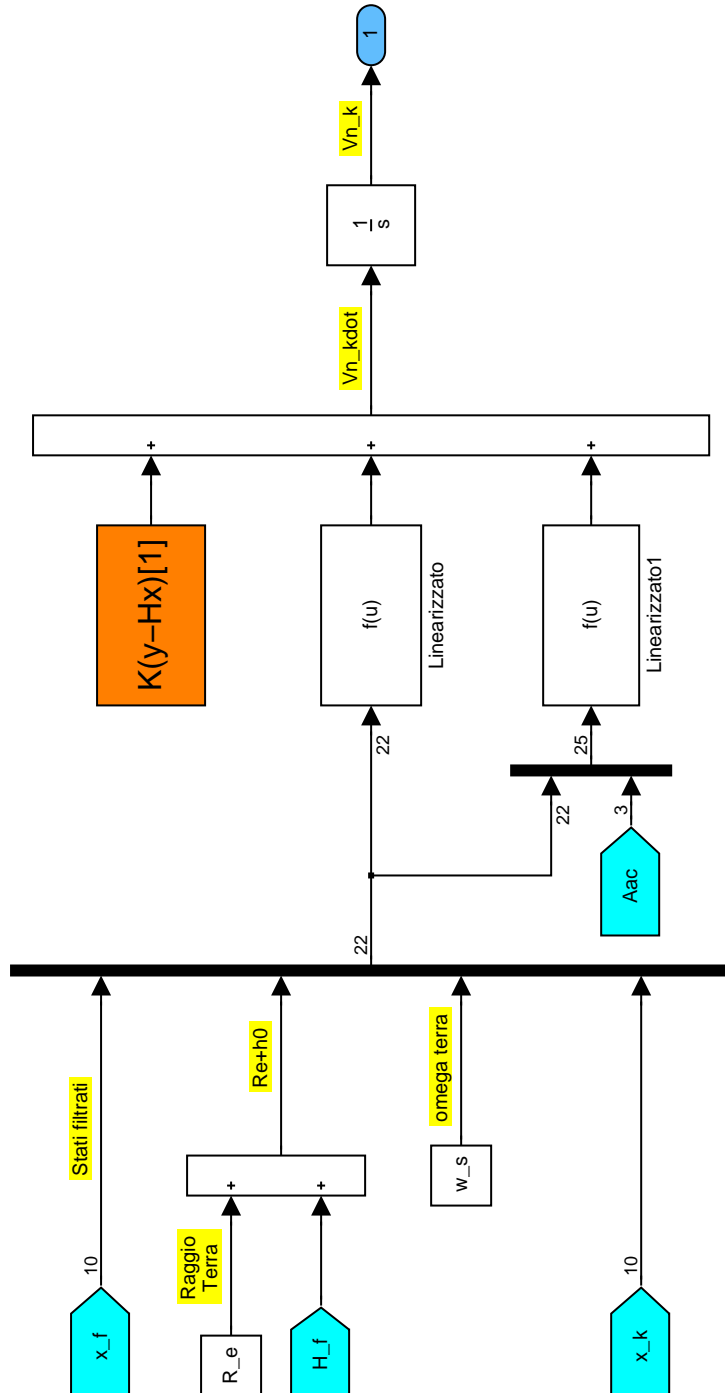


Figura 4.8: Blocco correzioni dello stato V_{Nord}

In particolare si nota un ulteriore blocco nel quale viene calcolato il termine correttivo di Kalman presente nell'equazione (4.16):

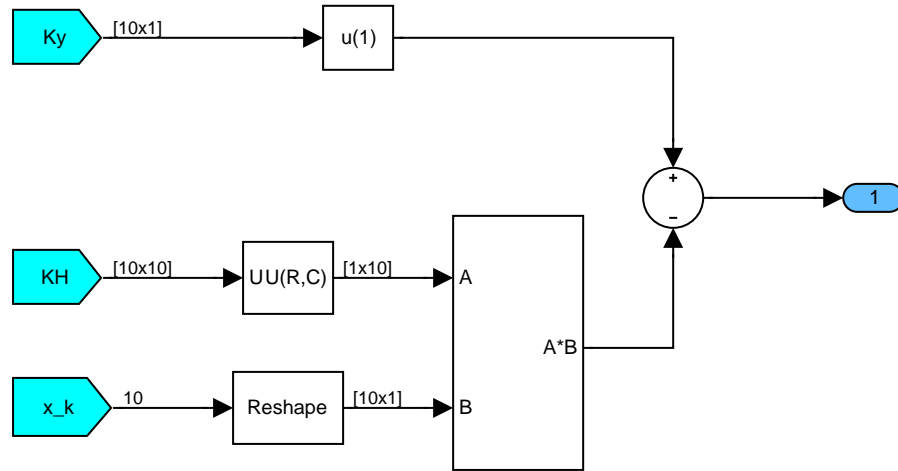


Figura 4.9: Blocco termine correttivo di Kalman

4.5 Esempi di simulazione

In analogia a quanto fatto a conclusione dei precedenti capitoli, si effettua ora una simulazione di test al fine di validare il modello di Kalman appena visto.

Si fa notare anticipatamente come i risultati relativi agli stati, riportati nelle seguenti figure, non debbano essere interpretati in termini di realtà fisica (essendo palesemente assurdi) ma come dimostrazione della validità del filtro nel ricostruire segnali del tutto generici.

Le condizioni iniziali di velocità, posizione ed assetto sono poste tutte pari a zero, le accelerazioni in ingresso sono delle sinusoidi con fase zero, frequenza pari a 0.1 rad/s e ampiezza di 5 m/s^2 mentre le velocità angolari sono identicamente nulle. Per la caratterizzazione del rumore si veda la Figura 4.10.

In tutte le figure si notano tre curve a confronto ed in particolare quella “ideale” è il risultato dell’integrazione del modello dinamico non lineare in assenza di rumore descritto nel capitolo 2. Per tale motivo essa viene presa come termine di paragone per le altre.

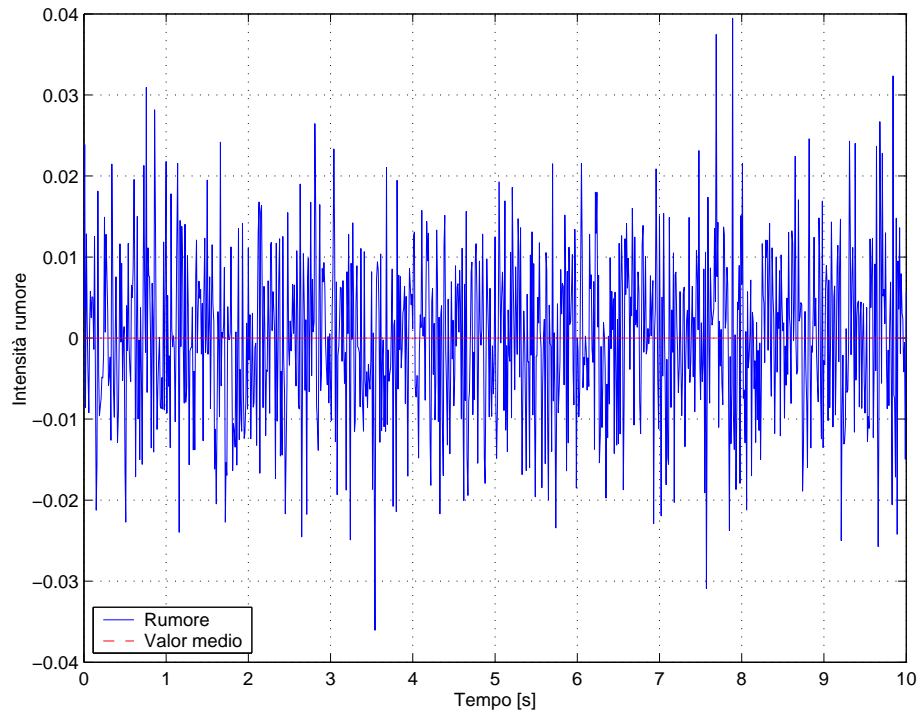


Figura 4.10: Entità del rumore utilizzato

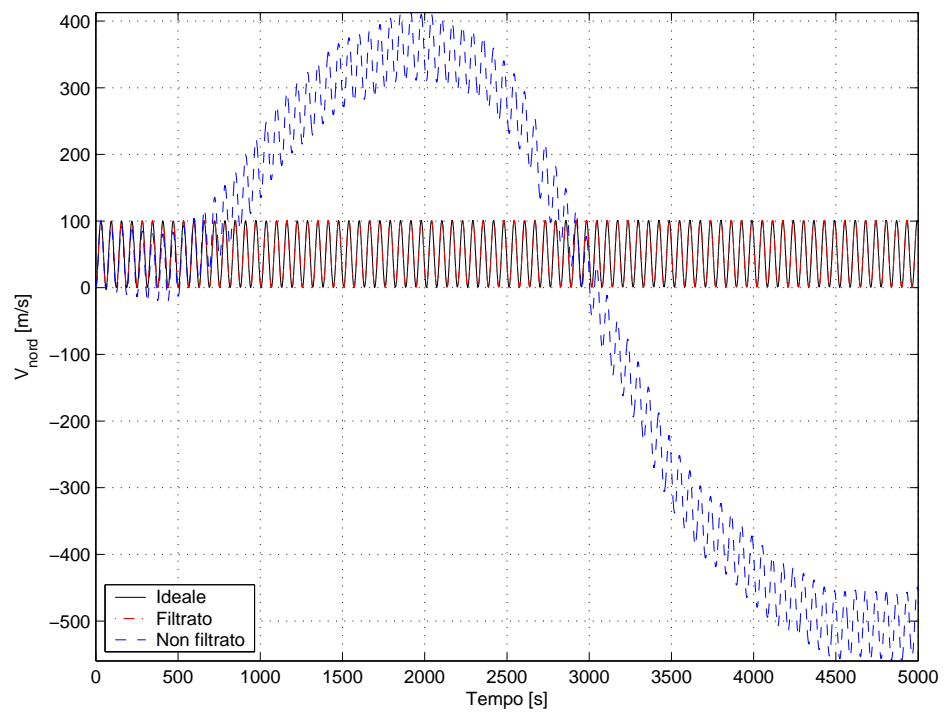


Figura 4.11: Confronto risultati simulazione per V_{Nord}

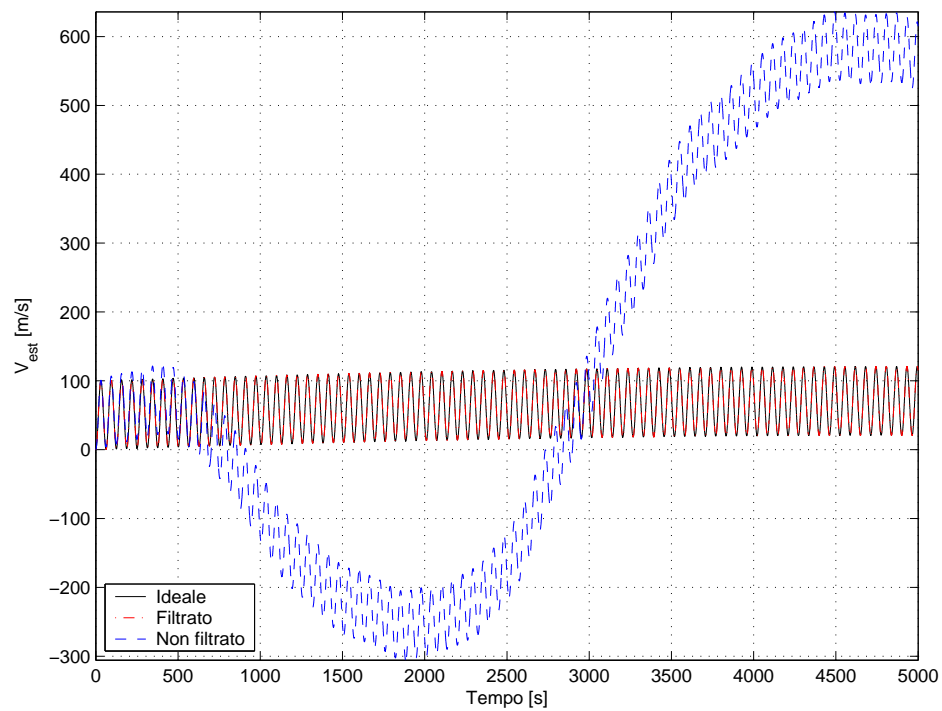


Figura 4.12: Confronto risultati simulazione per V_{Est}

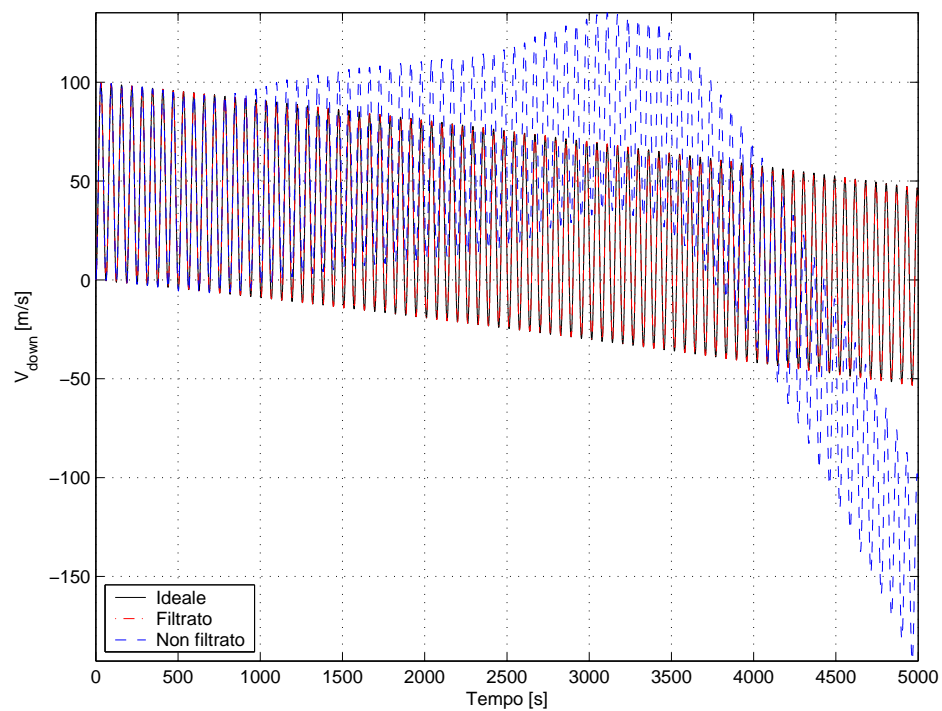


Figura 4.13: Confronto risultati simulazione per V_{Down}

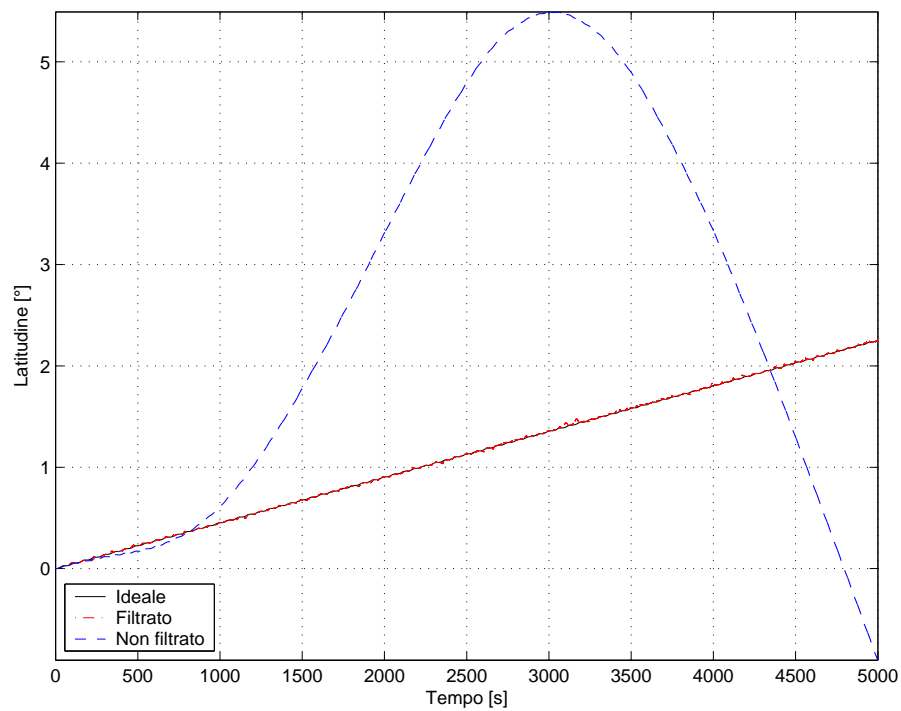


Figura 4.14: Confronto risultati simulazione per *Latitudine*

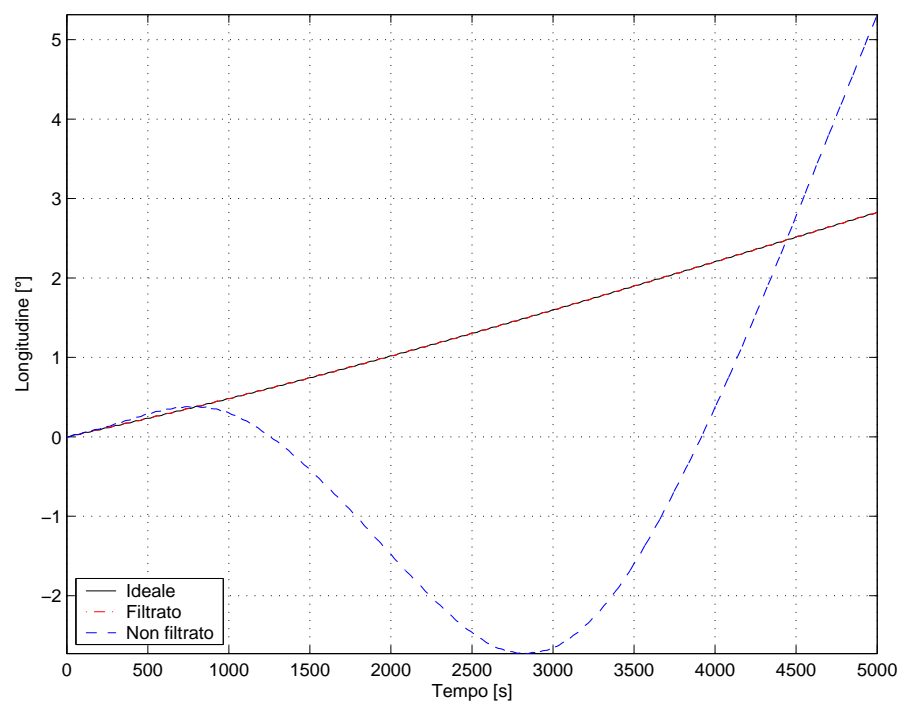


Figura 4.15: Confronto risultati simulazione per *Longitudine*

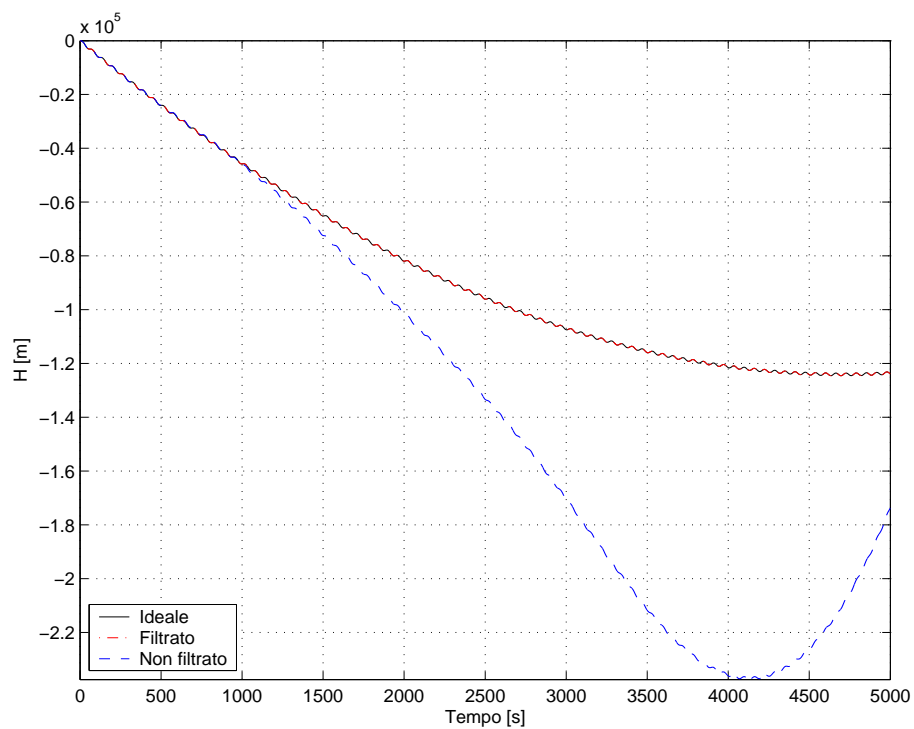


Figura 4.16: Confronto risultati simulazione per $Quota$

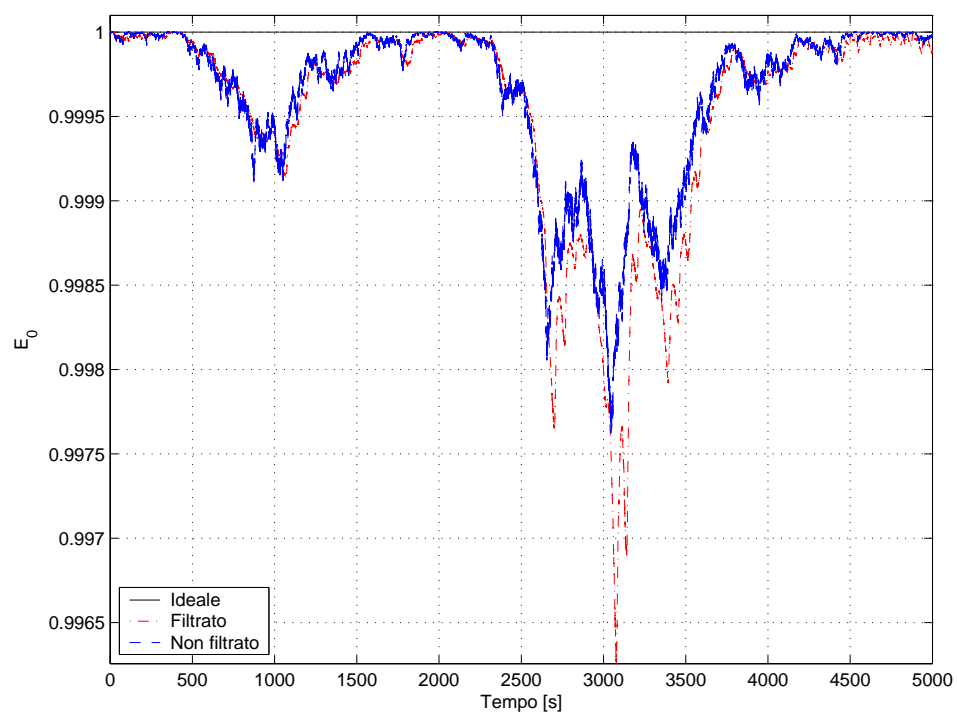


Figura 4.17: Confronto risultati simulazione per E_0

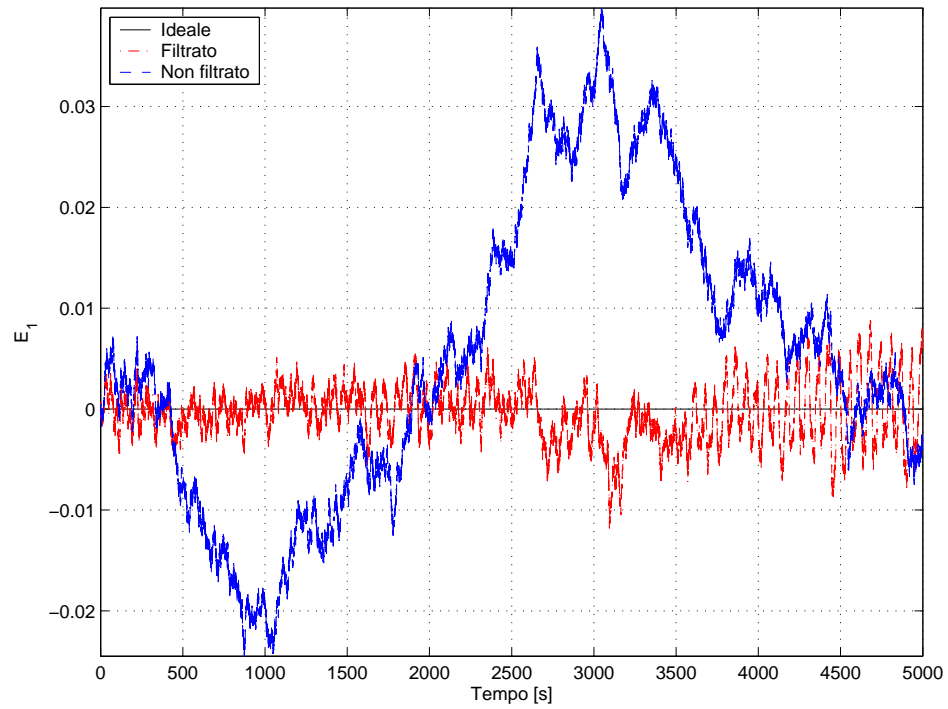


Figura 4.18: Confronto risultati simulazione per E_1

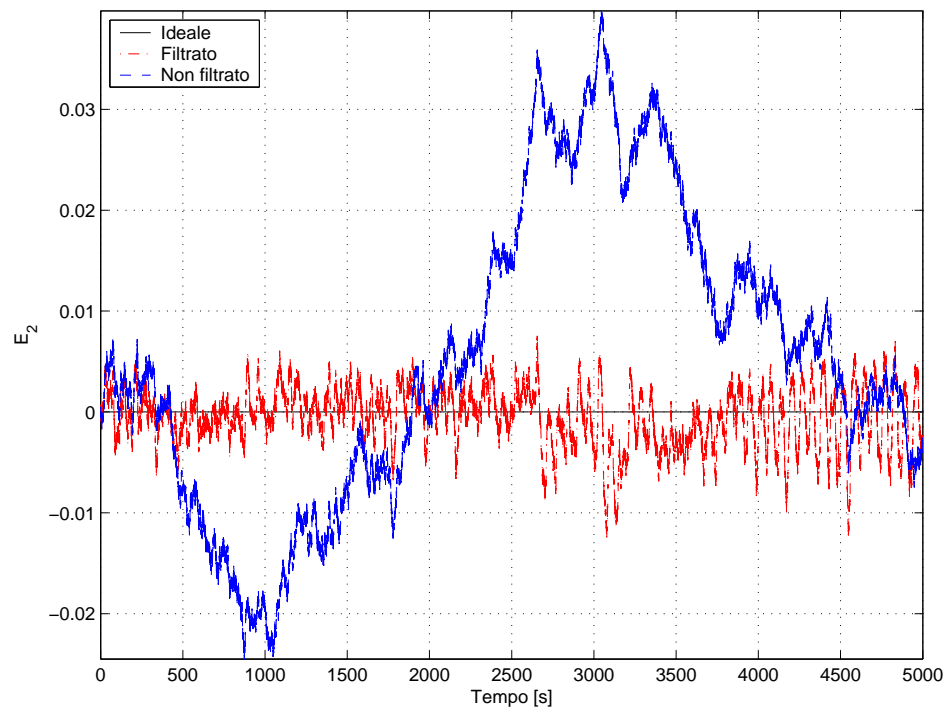


Figura 4.19: Confronto risultati simulazione per E_2

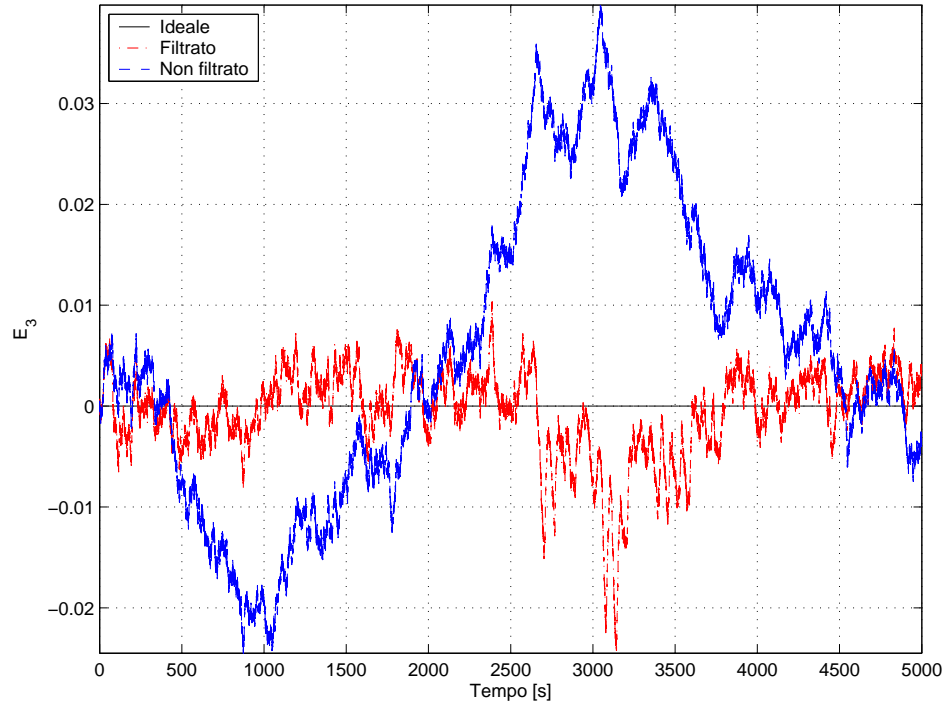


Figura 4.20: Confronto risultati simulazione per E_3

Dall’analisi delle figure si nota che gli stati corretti direttamente dalle misurazioni (le tre velocità, la posizione e la quota) seguono più fedelmente la traiettoria *Ideale* rispetto ai parametri di Eulero che subiscono invece una ricostruzione per così dire “indiretta”.

Probabilmente tale comportamento risulta ulteriormente accentuato dal fatto che la dinamica di assetto è totalmente disaccoppiata da quella del moto del baricentro.

Vale la pena di sottolineare che il passo di simulazione utilizzato in tutti gli esempi del presente lavoro è di $100Hz$ e che tale scelta è il risultato di un’attento studio di sensibilità sulla variazione dei risultati con il passo stesso.

Parte II

Risultati sperimentali

5

Attività sperimentale

5.1 Introduzione

Il presente capitolo ha lo scopo di illustrare le procedure per l'avvio del modello di simulazione e di mostrare i risultati dei test più significativi. A differenza degli esempi riportati nei precedenti capitoli, relativi a simulazioni numeriche, questi ultimi test verranno realizzati sperimentalmente mediante acquisizione dati dalla piattaforma inerziale e dal GPS in tempo reale (*Real Time*).

5.2 Preparazione dei test

5.2.1 Test con ingressi simulati

Al fine di garantire un corretto funzionamento del simulatore è necessario eseguire una serie di procedure nel seguente ordine:

1. Aprire i file “*LKF.mdl*” e “*Inizializzazione.m*”;
2. Modificare opportunamente il file “*Inizializzazione.m*” scegliendo le condizioni iniziali relative alla simulazione desiderata (velocità, posizione, quota, assetto, parametri *S-Function*, ecc.) e l'entità del rumore da assegnare alla variabile in esame;
3. Lanciare il file “*Inizializzazione.m*” mediante doppio clic sul pulsante “*Caricamento dati*” presente nel modello (vedi Figura 4.3);

4. Modificare gli ingressi e il tempo di simulazione;
5. Lanciare la simulazione.

5.2.2 Test “*Real Time*” con ingressi reali

I test di simulazione in “*Real Time*” vengono realizzati mediante l'utilizzo di due computer collegati tramite un cavo di rete ed aventi le seguenti funzioni:

- Il primo, detto *Server*, ha il compito di compilare il modello *Simulink* e di inviare l'eseguibile all'altro computer;
- Il secondo, detto *Target*, esegue il modello in tempo reale; esso, non avendo sistema operativo, effettua l'avvio da *hard-disk* ¹.

Come per il caso precedente elenchiamo le procedure per l'avvio della simulazione:

1. Accendere l'alimentatore che fornisce energia alla piattaforma inerziale e avvia il *Target*;
2. Eseguire i passi da 1-3 del §5.2.1 aprendo in questo caso il file denominato “*LKF_Target.mdl*”;
3. Verificare l'avvenuta connessione tra *Server* e *Target* digitando sulla *Command Window* di Matlab il comando “*xpctargetping*”;
4. Compilare il modello *Simulink* mediante il comando *Build* presente in *Simulation Parameters*;
5. Settare *Simulink* sul modo *external* e cliccare il tasto *Connect to Target*;
6. Lanciare la simulazione. ²

¹Per le istruzioni di settaggio di “*xPC Target*” si veda l'Appendice C.

²In alternativa ai passi 5 e 6 è possibile digitare il comando *+tg* per avviare la simulazione e *-tg* per arrestarla

5.3 Descrizione esperimento

L'esperimento di acquisizione dati in tempo reale si è svolto nel parcheggio del *Dipartimento di Ingegneria Aerospaziale* seguendo la seguente procedura:

- Per prima cosa è stata caricata sull'auto l'apparecchiatura *hardware* (vedi Figura 5.1) verificando il corretto allineamento degli assi sensibili della piattaforma inerziale con gli assi corpo della vettura;
- Dopo aver collocato l'antenna del ricevitore *GPS* sul tettuccio si è alimentato il tutto mediante una batteria;
- Una volta verificato il corretto funzionamento della strumentazione si è proceduto all'esecuzione dei passi 3÷6 del §5.2.2 compilando il *file* “*Salva_dati.mdl*” mediante un PC portatile.

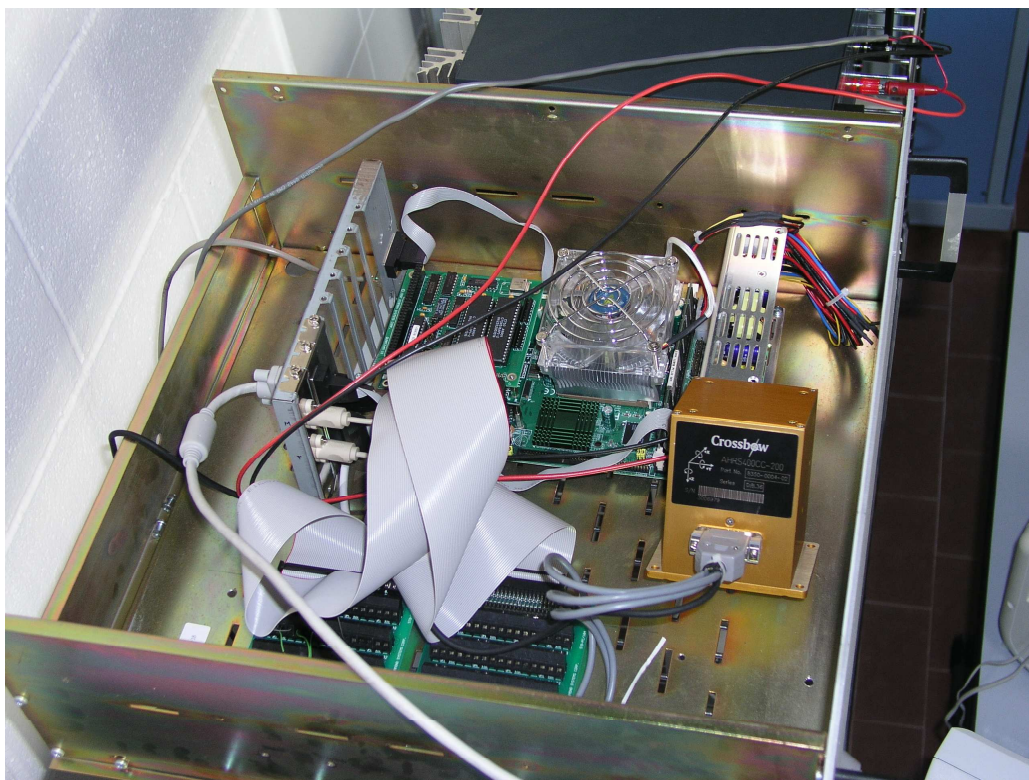


Figura 5.1: Foto strumentazione hardware

Il test più significativo si è protratto per 300s, durante i quali si sono memorizzati i segnali digitali provenienti dal *GPS* (ora, latitudine, longitudine, quota, velocità in assi verticali locali e numero di satelliti visibili) e dalla *piattaforma* (accelerazioni e velocità angolari in assi corpo, angoli di assetto e temperatura interna alla piattaforma).

Tali dati sono stati poi analizzati in laboratorio con operazione di *post-processing*. Quest'elaborazione ha permesso di confrontare la posizione stimata dal GPS con quella calcolata dalle misurazioni inerziali e di validare il filtro di Kalman per una situazione reale e non simulata.

5.4 Elaborazione dei dati

In quest'ultimo paragrafo si riportano i grafici più rappresentativi dell'elaborazione, ponendo particolare attenzione ai risultati corretti dal filtro di Kalman. Per avere un'idea della traiettoria seguita dall'autovettura durante il test, si riporta qui di seguito il grafico di *Longitudine-Latitudine*:

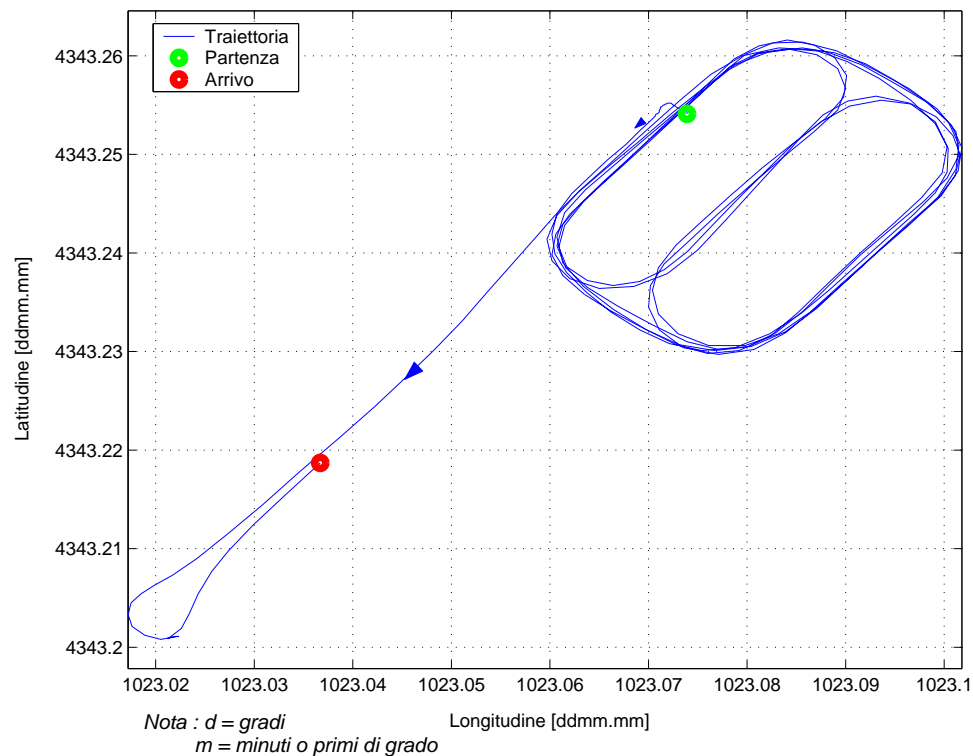


Figura 5.2: Traiettoria da acquisizione *GPS* in gradi

Al fine di avere una percezione più diretta degli spazi percorsi nel tragitto, si è riprodotta la medesima traiettoria di Figura 5.2 espressa però in metri:

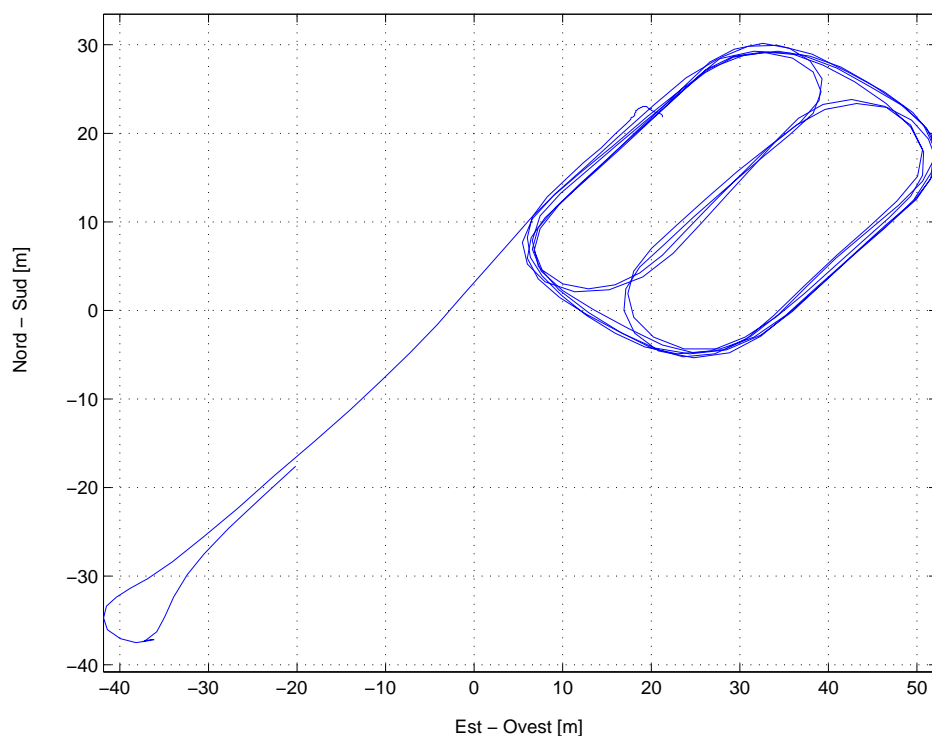


Figura 5.3: Traiettoria da acquisizione *GPS* in metri

Dalle figure che seguono si può notare con chiarezza come le correzioni del filtro di Kalman sulle grandezze derivate per integrazione dalle acquisizioni dalla piattaforma (velocità, posizione e assetto), siano indispensabili per una corretta navigazione inerziale. In particolare si capisce come gli errori sui segnali derivanti dalle accelerazioni (velocità e posizione) siano maggiori di quelli derivanti dai giroscopi (angoli di assetto).

Questo fatto è confermato dall'analisi della Figura 5.8 : si vede infatti che la curva “*IMU*” (angolo ψ calcolato e corretto mediante un filtro interno alla piattaforma) non si scosta molto dall'altra (ricavata per integrazione delle velocità angolari in assenza di filtraggio), a differenza di quanto accade ad esempio per la posizione (vedi Figura 5.6). Il risultato forse più significativo è evidenziato dalla Figura 5.9 : le due curve (relative alle acquisizioni da GPS e all'elaborazione del filtro) differiscono l'una dall'altra di pochi metri, precisione del tutto accettabile ai fini della navigazione.

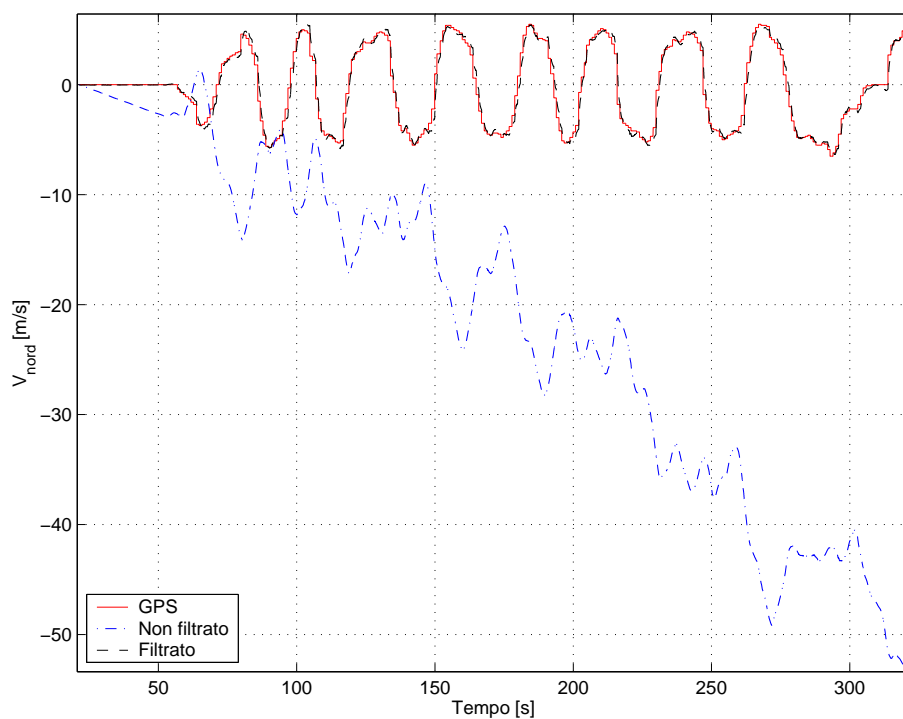


Figura 5.4: Confronto risultati per V_{Nord}

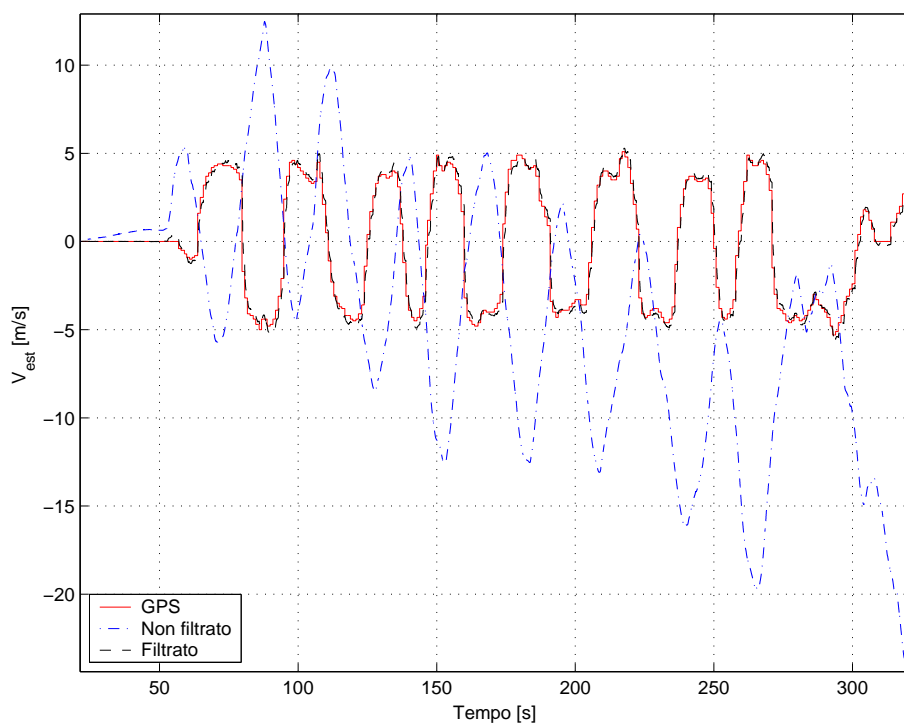


Figura 5.5: Confronto risultati per V_{Est}

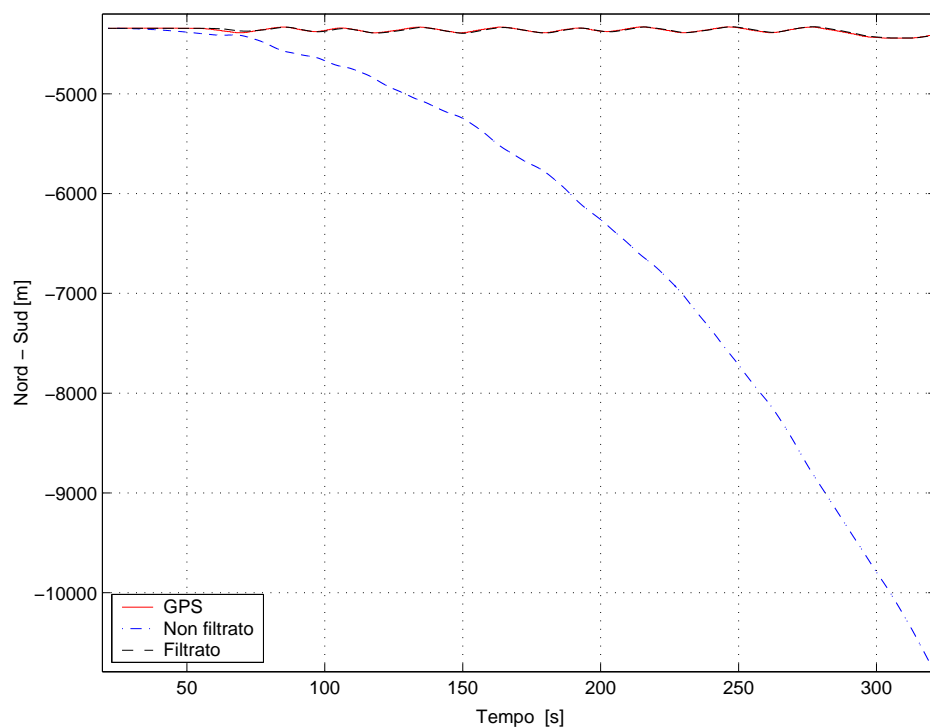


Figura 5.6: Confronto risultati per moto *Nord – Sud*

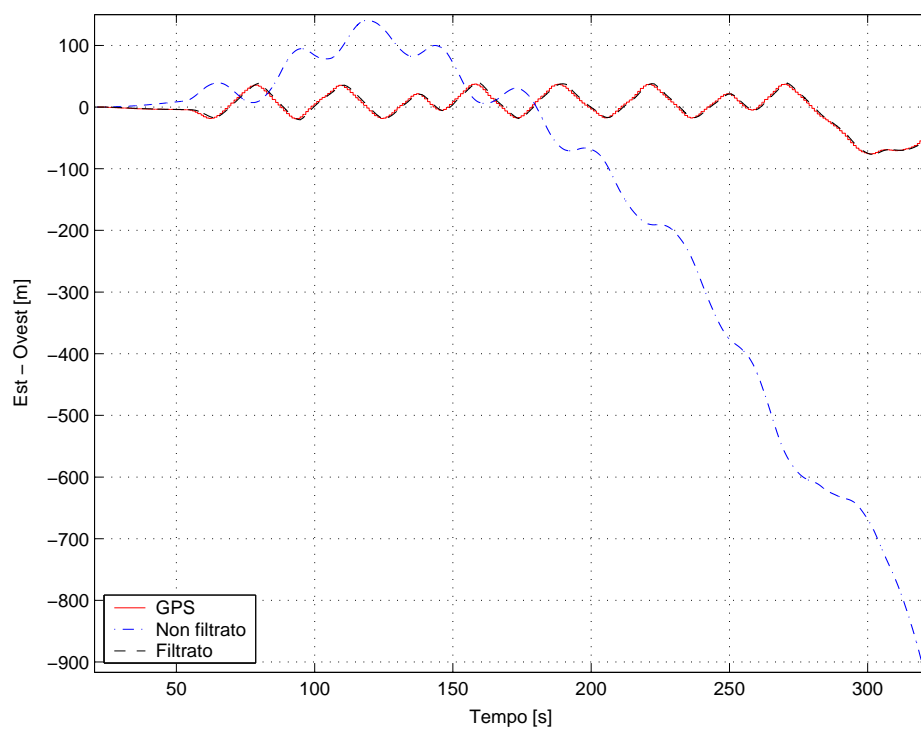


Figura 5.7: Confronto risultati per moto *Est – Ovest*

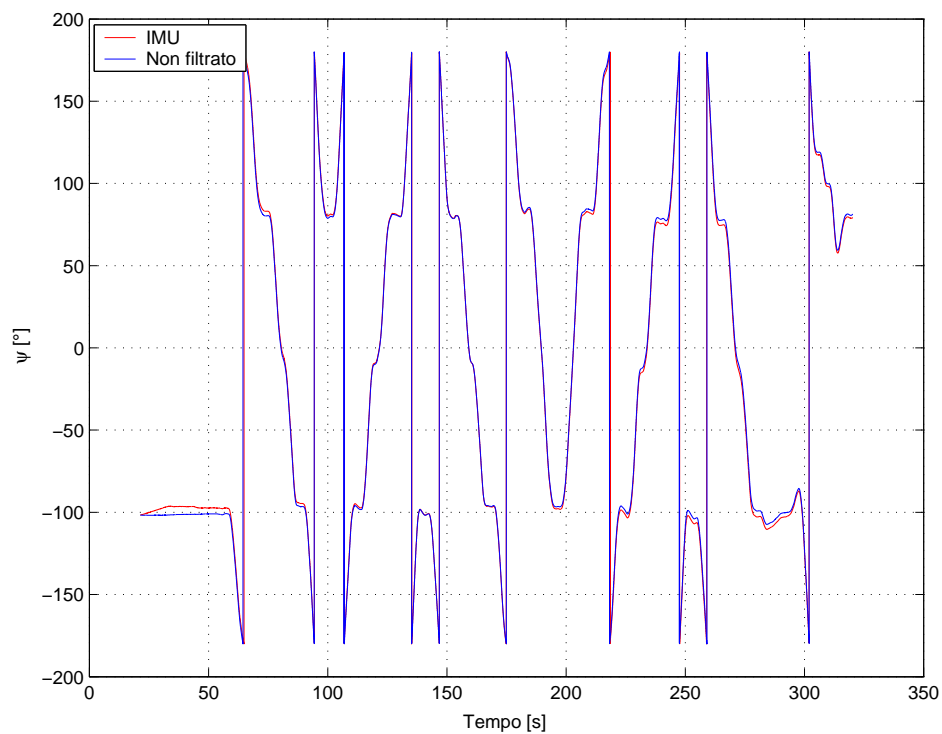


Figura 5.8: Confronto risultati angolo imbardata ψ

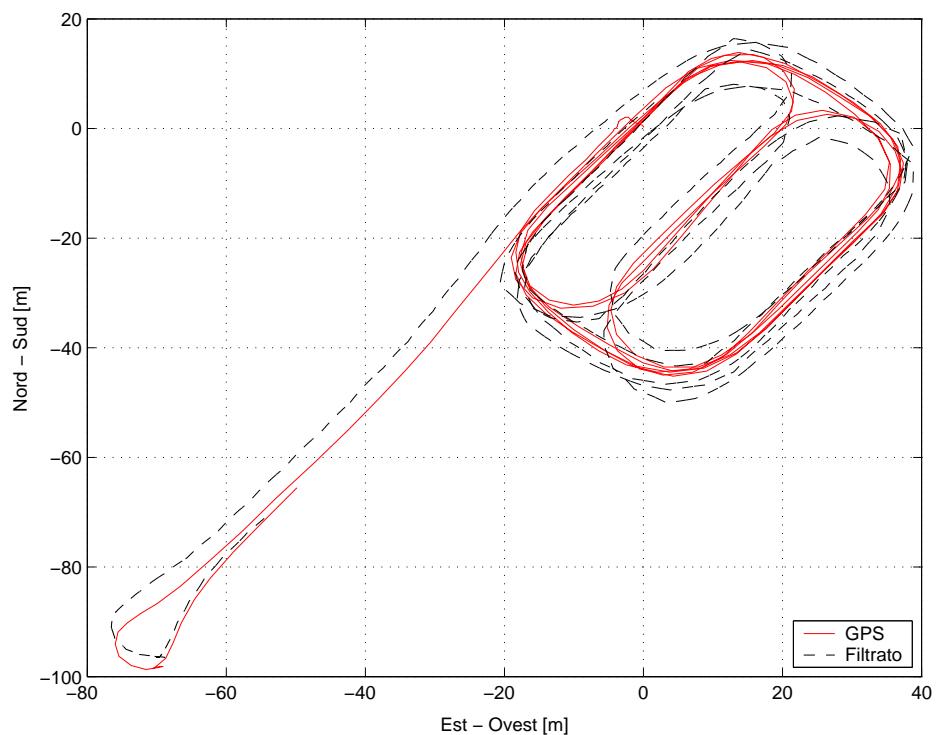


Figura 5.9: Confronto fra $Traccia_{(GPS)} - Traccia_{(INS)}$

6

Conclusioni e sviluppi futuri

6.1 Conclusioni

Quanto esposto fin'ora permette di asserire che gli obiettivi di questa tesi sono stati raggiunti:

1. Il primo obiettivo è stato quello di programmare in Simulink un software per la navigazione inerziale di un velivolo (*Inertial Navigation System*) ed un filtro di Kalman ad esso applicato, realizzato tramite integrazione di dati fra GPS ed INS;
2. Il secondo quello di progettare e realizzare una strumentazione hardware per l'acquisizione e l'elaborazione in tempo reale dei dati provenienti dalla piattaforma inerziale e dal GPS;
3. In ultimo si sono uniti il software e l'hardware al fine di procedere con i test sperimentali di validazione.

6.2 Sviluppi futuri

La continuazione di questo lavoro prevede un ulteriore affinamento del filtro di Kalman (introducendo ad esempio le misurazioni degli angoli di assetto per mezzo di un qualche strumento diverso dalla piattaforma) e la sostituzione dell'attuale piattaforma integrata (troppo pesante e sofisticata per un UAV) con accelerometri e giroscopi meno costosi (e meno accurati).

I prossimi esperimenti dovranno inoltre includere il software di elaborazione

dati e non solo quello di acquisizione, in previsione questo dei futuri test di volo dell'UAV.

Appendice **A**

Linearizzazione

Un qualunque sistema non lineare può essere scritto nella seguente forma funzionale:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \quad (\text{A.1})$$

dove

- \mathbf{f} Funzione nota, rappresentante la dinamica del sistema fisico
- \mathbf{x} Vettore degli stati del sistema
- \mathbf{u} Vettore degli ingressi al sistema

Supponiamo ora che esista una traiettoria di riferimento “ideale” rispetto alla quale andremo a linearizzare. Imponiamo perciò che la traiettoria “reale” sia data da:

$$\mathbf{x}(t) = \mathbf{x}^*(t) + \Delta\mathbf{x}(t) \quad (\text{A.2})$$

dove

- $\mathbf{x}^*(t)$ Vettore degli stati nominale o di riferimento
- $\Delta\mathbf{x}(t)$ Vettore degli incrementi

Sostituendo l'equazione (A.2) nella (A.1) otteniamo:

$$\dot{\mathbf{x}}^* + \Delta\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}^* + \Delta\mathbf{x}, \mathbf{u}, t) \quad (\text{A.3})$$

Supponendo che tali incrementi siano piccoli è possibile sviluppare quest'ultima espressione in serie di *Taylor* arrestata al primo ordine nell'intorno della condizione di equilibrio:

$$\Delta \dot{\mathbf{x}} = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right]_{\mathbf{x}=\mathbf{x}^*} \cdot \Delta \mathbf{x} + \left[\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right]_{\mathbf{u}=\mathbf{u}^*} \cdot \Delta \mathbf{u} \quad (\text{A.4})$$

dove

$$\mathbf{A} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad (\text{A.5})$$

$$\mathbf{B} = \frac{\partial \mathbf{f}}{\partial \mathbf{u}} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \dots & \frac{\partial f_1}{\partial u_m} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \dots & \frac{\partial f_2}{\partial u_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \frac{\partial f_n}{\partial u_2} & \dots & \frac{\partial f_n}{\partial u_m} \end{bmatrix} \quad (\text{A.6})$$

e dove abbiamo implicitamente assunto che:

$$\dot{\mathbf{x}}^* = \mathbf{f}(\mathbf{x}^*, \mathbf{u}, t) \quad (\text{A.7})$$

L'equazione (A.4) riscritta in forma classica diventa:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (\text{A.8})$$

Nella linearizzazione del filtro di Kalman troviamo anche lo jacobiano associato alle misurazioni \mathbf{H} [vedi eq. (4.11)] che, in analogia ad \mathbf{A} e \mathbf{B} , diventa:

$$\mathbf{H} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \cdots & \frac{\partial h_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_r}{\partial x_1} & \frac{\partial h_r}{\partial x_1} & \cdots & \frac{\partial h_r}{\partial x_n} \end{bmatrix} \quad (\text{A.9})$$

nella quale $\mathbf{h}(\mathbf{x})$ è una funzione nota che lega le misurazioni agli stati del sistema.

Approfondimento sulla S-Function

La *S-Function* nel modello *Simulink* relativo al filtro di Kalman (vedi Figura 4.7), ha lo scopo di integrare le $\dot{\mathbf{P}}$ definite da (4.23) e calcolare la matrice dei guadagni di Kalman \mathbf{K} (4.20).

L'utilizzo di tale strumento, dato l'elevato numero di equazioni, è risultato indispensabile: la matrice \mathbf{P} ha infatti dimensione $[10 \times 10]$ e anche se viene forzata ad essere simmetrica per motivi di stabilità numerica [1], rimangono comunque 55 equazioni differenziali indipendenti da risolvere (gli elementi della relativa matrice triangolare). In questo modo è stato possibile snellire notevolmente il modello Simulink rendendolo più leggibile.

Una *S-Function* è composta sostanzialmente da quattro sottofunzioni principali definite come segue:

“**mdlInitializeSizes**” in cui vengono specificate le dimensioni dei vettori di ingresso, di uscita, degli stati e il numero dei parametri;

“**mdlInitializeConditions**” in cui vengono definite le condizioni iniziali di integrazione;

“**mdlOutputs**” in cui vengono calcolate le uscite;

“**mdlDerivatives**” in cui si definiscono le derivate relative alle equazioni differenziali.

Gli ingressi e le uscite di questo lavoro sono indicate in Figura 4.7 mentre i parametri sono elencati all'interno del *block parameters* dell'*S-Function* di Figura 4.7. In particolare sono parametrizzati gli elementi delle matrici di

covarianza degli stati e delle misurazioni (\mathbf{v} e \mathbf{w}) e le condizioni iniziali di integrazioni della matrice di covarianza dell'errore (\mathbf{P}_0). Tali valori vengono introdotti attraverso il file di inizio simulazione “*Inizializzazione.m*”.

Settaggio di “*xPC Target*”

Lo scopo principale di quest’appendice è quello di fornire le procedure di configurazione di un PC, al fine di poter effettuare simulazioni in tempo reale mediante xPC Target.

- Installare sul PC *Sorgente* (nel quale si trova il modello Simulink da compilare) il Visual C++ contenuto all’interno del software MS-Visual Studio 6.0. Tra le impostazioni proposte selezionare la voce *Registrare variabili d’ambiente*;
- Avviare il Matlab sul computer *Sorgente* e sulla Command Window digitare il comando *xpcsetup*;
- Il precedente comando provoca l’apertura di una finestra di dialogo (vedi Figura C.1), i cui campi vanno configurati come segue:
 - selezionare la voce Visual C nella casella *CCcompiler*;
 - inserire nella casella *Compiler Path* il percorso della directory dove è stato installato il Visual C++;
 - selezionare la voce TCP/IP nella casella *Host Target Comm*;
 - inserire nelle caselle *TcpIp Target Address* , *TcpIp Target Port*, *TcpIp Subnet Mask* e *TcpIp Gateway* le omonime stringhe di numeri presenti nella schermata di avvio del monitor attaccato al *Target* (vedi Figura C.2);

- inserire il tipo di scheda di acquisizione utilizzata nella casella *TcpIp Target Driver* (in questo lavoro sono state utilizzate schede provviste di chip *I82559*);

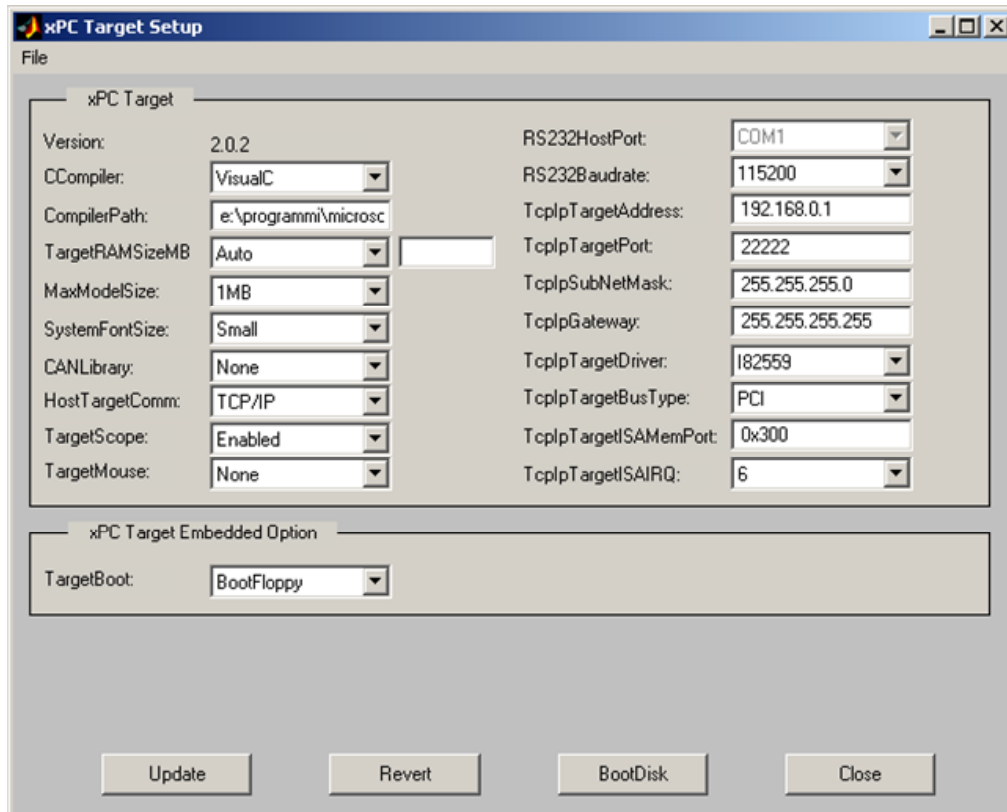


Figura C.1: Finestra di configurazione per *xPC Target*

- Cliccare sul pulsante **Update** per rendere esecutive le impostazioni del punto precedente;
- Avviare il PC del *Target* il cui monitor mostrerà la finestra di Figura C.2;
- Per verificare se le configurazioni precedenti sono esatte digitare sulla Command Window di Matlab il comando `xpctest` ed aspettare l'esito favorevole evidenziato dalla scritta “*Test Suite successfully finished*”;
- Sul computer *Sorgente* aprire il modello Simulink da compilare.
Nota : eliminare dal modello, se è presente, il blocco *clock*;

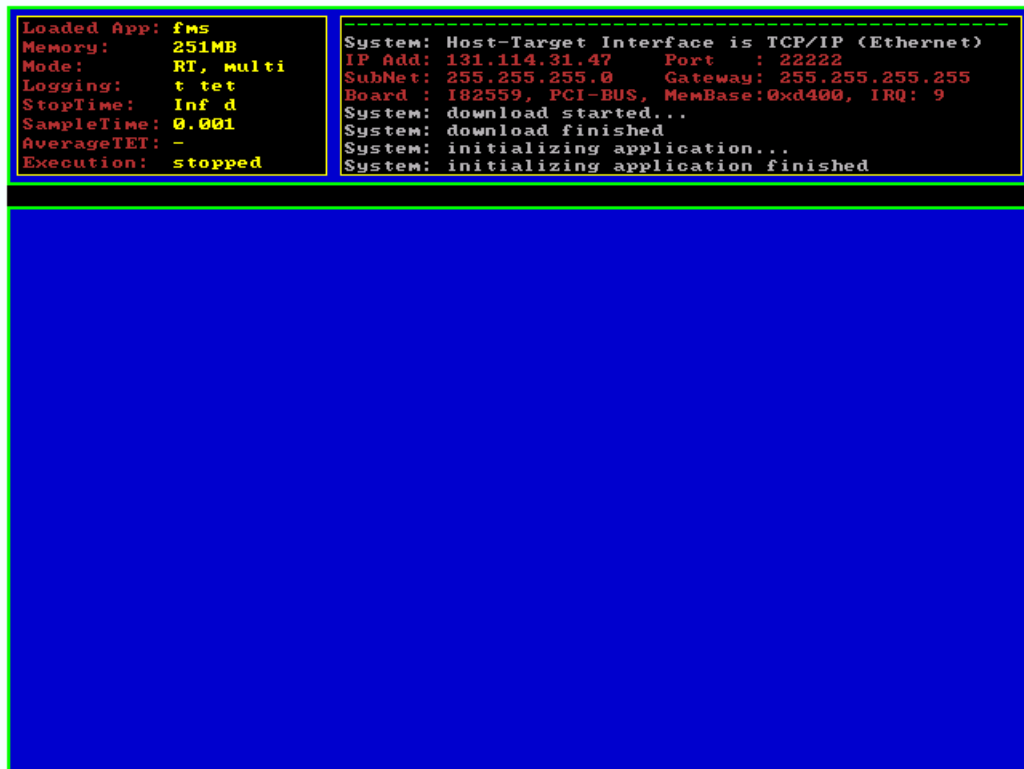


Figura C.2: Finestra di avvio di *xPC Target*

- Digitare *Ctrl+E* e all’apertura pannello *Simulation Parameters* eseguire quanto segue:
 - selezionare nel campo *Category* la voce *Target configuration*;
 - mediante il tasto Browse e scegliere come *System target file* la voce *xpctarget.tlc*;
 - cliccare sul tasto Build ; questo comando determina la compilazione del modello Simulink relativo e l’automatica connessione al *Target*;
- Avviare la simulazione dopo aver settato il modo *External*.

Bibliografia

- [1] Brown, R. G., *Introduction to Random Signals and Applied Kalman Filtering*, John Wiley & sons, inc., 1983.
- [2] Casarosa, C., “Meccanica del Volo,” Dispense del corso di Meccanica del Volo, Corso di Laurea in Ingegneria Aerospaziale A.A. 2002.
- [3] Farrel, J. L., *Integrated Aircraft Navigation*, Academic press, inc., 111, Fifth Avenue, New York, 10003, 1976.
- [4] Cooke, J. M., Zyda, M. J., Pratt, D. R., e McGhee, R. B., “NPS-NET:Flight Simulation Dynamic Modeling Using Quaternion,” Atti di dipartimento, Naval Postgraduate School, Department of Computer Science, Monterey, California 93943-5100 USA, January 25 1994.
- [5] Ogata, K., *Modern Control Engineering*, Prentice Hall International ,inc., Englewood Cliffs, New Jersey, 07632, 1990.

Elenco delle figure

1.1	Esempio di <i>rumore bianco</i>	2
1.2	Esempio di integrazione di <i>rumore bianco</i>	3
1.3	Schema dell'architettura del filtro	5
1.4	Schema dell'architettura hardware	6
1.5	Schema elettronico AHRS	7
2.1	Definizione dei quaternioni	15
2.2	Blocco principale modello non lineare	17
2.3	Blocco ingressi accelerazioni	18
2.4	Blocco accelerometri	19
2.5	Limitazione longitudine a $+180^\circ$	20
2.6	Limitazione longitudine a -180°	20
2.7	Blocco phi	21
2.8	Blocco Nord	22
2.9	Blocco ingressi velocità angolari	23
2.10	Blocco giroscopi	24
2.11	Simulazione con modello non lineare	26
3.1	Blocco principale modello linearizzato	31
3.2	Blocco vettore degli stati	32
3.3	Blocco vettore incrementi degli stati	33
3.4	Blocco vettore incrementi delle velocità angolari	34
3.5	Blocco vettore incrementi delle accelerazioni	34
3.6	Blocco ingressi accelerazioni	35
3.7	Blocco ingressi velocità angolari	35
3.8	Blocco Nord	36

3.9	Blocco Parametri di Eulero (E_0)	37
3.10	Esempi di simulazione a confronto	40
3.11	Dettaglio della zona di scostamento	41
4.1	Schema algoritmo di calcolo filtro di Kalman	43
4.2	Schema dell'osservatore (filtro di Kalman)	47
4.3	Blocco principale modello completo	49
4.4	Ingressi accelerazioni INS con rumore	50
4.5	Blocco di campionamento segnali	51
4.6	Blocco filtro di Kalman	51
4.7	Blocco calcolo matrici "H" e "K"	52
4.8	Blocco correzioni dello stato V_{Nord}	53
4.9	Blocco termine correttivo di Kalman	54
4.10	Entità del rumore utilizzato	55
4.11	Confronto risultati simulazione per V_{Nord}	55
4.12	Confronto risultati simulazione per V_{Est}	56
4.13	Confronto risultati simulazione per V_{Down}	56
4.14	Confronto risultati simulazione per <i>Latitudine</i>	57
4.15	Confronto risultati simulazione per <i>Longitudine</i>	57
4.16	Confronto risultati simulazione per <i>Quota</i>	58
4.17	Confronto risultati simulazione per E_0	58
4.18	Confronto risultati simulazione per E_1	59
4.19	Confronto risultati simulazione per E_2	59
4.20	Confronto risultati simulazione per E_3	60
5.1	Foto strumentazione hardware	64
5.2	Traiettoria da acquisizione <i>GPS</i> in gradi	65
5.3	Traiettoria da acquisizione <i>GPS</i> in metri	66
5.4	Confronto risultati per V_{Nord}	67
5.5	Confronto risultati per V_{Est}	67
5.6	Confronto risultati per moto <i>Nord – Sud</i>	68
5.7	Confronto risultati per moto <i>Est – Ovest</i>	68
5.8	Confronto risultati angolo imbardata ψ	69
5.9	Confronto fra $Traccia_{(GPS)} - Traccia_{(INS)}$	69
C.1	Finestra di configurazione per <i>xPC Target</i>	78

C.2 Finestra di avvio di <i>xPC Target</i>	79
--	----

Elenco degli acronimi

SCAUT Sistema di Controllo AUtomatico del Territorio

DIA Dipartimento di Ingegneria Aerospaziale

DIASP Dipartimento di Ingegneria Aeronautica e Spaziale del Politecnico di Torino

UAV *Uninhabited Aerial Vehicle*

PC Personal Computer

AHRS Attitude & Heading Reference System

INS *Inertial Navigation System*

MEMS Micro Electro Mechanical Systems

GPS *Global Position System*